



Titre: Vérification de la méthode de Boltzmann sur réseau en vue de
Title: calculer la perméabilité de milieux poreux

Auteur: Maxence Neveu
Author:

Date: 2017

Type: Mémoire ou thèse / Dissertation or Thesis

Référence: Neveu, M. (2017). Vérification de la méthode de Boltzmann sur réseau en vue de
Citation: calculer la perméabilité de milieux poreux [Mémoire de maîtrise, École
Polytechnique de Montréal]. PolyPublie. <https://publications.polymtl.ca/2748/>

 **Document en libre accès dans PolyPublie**
Open Access document in PolyPublie

URL de PolyPublie: <https://publications.polymtl.ca/2748/>
PolyPublie URL:

Directeurs de recherche: Marcelo Reggio, & Sebastien Leclaire
Advisors:

Programme: Génie mécanique
Program:

UNIVERSITÉ DE MONTRÉAL

VÉRIFICATION DE LA MÉTHODE DE BOLTZMANN SUR RÉSEAU EN VUE DE
CALCULER LA PERMÉABILITÉ DE MILIEUX POREUX

MAXENCE NEVEU
DÉPARTEMENT DE GÉNIE MÉCANIQUE
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION
DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES
(GÉNIE MÉCANIQUE)
AOÛT 2017

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé :

VÉRIFICATION DE LA MÉTHODE DE BOLTZMANN SUR RÉSEAU EN VUE DE
CALCULER LA PERMÉABILITÉ DE MILIEUX POREUX

présenté par : NEVEU Maxence

en vue de l'obtention du diplôme de : Maîtrise ès sciences appliquées

a été dûment accepté par le jury d'examen constitué de :

M. TRÉPANIÉ Jean-Yves, Ph. D., président

M. REGGIO Marcelo, Ph. D., membre et directeur de recherche

M. LECLAIRE Sébastien, Ph. D., membre et codirecteur de recherche

M. AMMAR Sami, Ph. D., membre

REMERCIEMENTS

Je tiens à remercier Marcelo Reggio, sans qui je n'aurais probablement jamais effectué une maîtrise en recherche. Ta grande confiance, depuis le tout début, était un élément clé à la réalisation de mon mémoire. Ta patience et ta disponibilité ont mené à de longues discussions durant lesquelles j'ai pu avoir une perspective pertinente sur le petit monde de la recherche sur la méthode de Boltzmann sur réseau, et le grand monde de la recherche en général.

Je tiens également à remercier Sébastien Leclaire, sans qui le niveau technique de mon mémoire n'aurait jamais été aussi élevé. Ton soutien et ton ouverture ont mené à des échanges captivants concernant la méthode de Boltzmann sur réseau. Tu as su me transmettre une passion et un sens de la rigueur scientifique qui m'ont poussé à me dépasser et ainsi produire un mémoire dont je peux aujourd'hui être fier.

Merci à ma conjointe Catherine Vaillancourt qui a su m'aider et me supporter dans la réalisation de ce mémoire. Durant cette période, nous avons pu ensemble partager les joies et les difficultés de l'initiation au monde de la recherche. Ton soutien inconditionnel durant cette épreuve est encourageant pour les prochains défis que nous aurons à relever.

Merci à ma famille et à mes amis qui ont su m'encourager et m'inspirer avant et pendant la réalisation de ce mémoire.

RÉSUMÉ

Le présent mémoire aborde plusieurs aspects de la méthode de Boltzmann sur réseau (*Lattice Boltzmann Method*), de la théorie à l'application. En premier lieu, le cadre théorique entourant les équations employées par cette méthode est abordé afin de mettre en contexte leur signification physique. En ce sens, la théorie cinétique des gaz est brièvement abordée, puis un système d'équations intégro-différentielles est présenté afin de décrire le comportement microscopique des particules d'un écoulement. Ces équations sont composées de l'équation de Boltzmann ainsi que de l'approximation BGK (Bhatnagar–Gross–Krook) pour l'effet des collisions entre particules.

Les différentes étapes de discrétisation menant d'un système d'équations continues à un système d'équations discrétisées sont présentées afin d'aboutir à la méthode de Boltzmann sur réseau, sous sa forme la plus simple. Cette forme permet de simuler la dynamique d'écoulements incompressibles, instationnaires, isothermes et subsoniques. Ensuite, la librairie C++ Palabos est employée afin de profiter d'une structure de programmation orientée objet et d'une implémentation optimisée pour le calcul en parallèle sur plusieurs processeurs.

L'application de la méthode LB visée est la simulation de l'infiltration d'eau dans les sols granulaires. Ces sols sont considérés comme des milieux poreux et leur comportement est modélisé par la perméabilité définie selon la loi de Darcy. Le but éventuel serait donc de calculer numériquement la perméabilité de ces milieux poreux en fonction de divers paramètres, tels que leur géométrie et leur teneur en eau. On profiterait alors de la capacité de la méthode LB à simuler des écoulements en milieux poreux afin d'améliorer les modèles hydrologiques macroscopiques à partir de simulations au niveau microscopique. Le type d'écoulement concerné est un écoulement diphasique immiscible (air et eau) à travers un milieu poreux. Dans le présent mémoire, on s'intéresse au cas plus simple constitué d'un écoulement à une seule phase (eau), ce qui correspond au cas particulier de sol saturé en eau. L'écoulement est simulé à travers un milieu granulaire idéalisé composé d'un réseau structuré de sphères.

Plusieurs cas tests sont abordés afin de vérifier la méthode LB et d'observer l'effet de divers paramètres sur la précision, la stabilité et le temps de calcul. Les cas tests sont l'écoulement de Poiseuille en 2D et 3D, la cavité carrée 2D avec paroi entraînée, l'écoulement de Poiseuille 2D avec obstacle cylindrique, et un écoulement 3D à travers un réseau de sphères. Les paramètres étudiés sont entre autres : le nombre de Reynolds, les pas de discrétisation temporelle et spatiale, les conditions aux frontières, les opérateurs de collision. Certains des effets numériques abordés sont : la compressibilité de la méthode, la non-conservation de la masse pour

certaines conditions aux frontières, les effets de la discrétisation géométrique du maillage. Les écoulements de Poiseuille 2D et 3D sont vérifiés à partir de la solution analytique, alors que la force de traînée sur l'obstacle cylindrique ou la perméabilité de réseaux de sphères sont vérifiées en comparant avec des résultats de la littérature.

Une analyse des erreurs pour le calcul de la perméabilité de milieux poreux, utilisant la frontière *bounce-back*, est effectuée sur les deux derniers cas tests. On conclut entre autres que l'erreur est dominée par l'effet de la discrétisation de la géométrie ; et que le modèle MRT (Multiple-Relaxation-Time) semble avantageux pour sa plus faible sensibilité à la résolution temporelle.

ABSTRACT

This paper discusses several aspects of the Lattice Boltzmann Method (LBM), from theory to application. The theoretical framework surrounding the equations used by this method is first introduced in order to put into context their physical meaning. In this sense, the kinetic theory of gases is briefly discussed, then a system of integro-differential equations is presented to describe the microscopic behavior of the particles of a flow. This system is composed of the Boltzmann equation as well as the BGK (Bhatnagar–Gross–Krook) approximation for the effect of the collisions between particles.

The different discretization steps leading from a system of continuous equations to a system of discretized equations are presented in order to recover the simplest form of the Lattice Boltzmann Method. This form allows to simulate the dynamics of incompressible, unsteady, isothermal and subsonic flows. Then, the Palabos C++ library is used to take advantage of an object-oriented programming structure and an optimized implementation for parallel computing.

The anticipated application of LBM is the simulation of water infiltration in granular soils. These soils are considered as porous media and their behavior is modeled by the permeability defined by the Darcy law. The aim would be to numerically calculate the permeability of these porous media as a function of various parameters, such as their geometry and water content. The ability of the LBM to simulate flows in porous media could be used to improve macroscopic hydrological models from simulations at the microscopic level. The type of flow involved is a two-phase flow (air and water) through a porous medium. In the present paper, we are interested in the simpler case consisting of a single-phase flow (water), which corresponds to the particular case of soil saturated with water. The flow is computed through an idealized granular medium composed of a structured arrangement of spheres.

Several test cases are considered in order to verify the LB method and observe the effect of various parameters on accuracy, stability and computation time. The test cases are Poiseuille flow in 2D and 3D, 2D lid driven cavity, 2D Poiseuille flow with cylindrical obstacle, and 3D flow through a network of spheres. Some of the parameters studied are : the Reynolds number, temporal and spatial discretization steps, boundary conditions, collision operators. Some of the numerical effects discussed are : the compressibility of the method, the non-conservation of the mass of certain boundary conditions, the effects of geometric discretization of the mesh. The Poiseuille 2D and 3D flows are verified with the analytical solution, while the drag force on a cylindrical obstacle and the permeability of sphere packings are verified

by comparing with results from the literature.

An error analysis for the calculation of the permeability of porous media, using the *bounce-back* boundary, is performed on the last two test cases. We conclude that error is dominated by the effect of discretization of geometry; and the MRT (Multiple-Relaxation-Time) model seems to be advantageous for its lower sensitivity to the time step.

TABLE DES MATIÈRES

REMERCIEMENTS	iii
RÉSUMÉ	iv
ABSTRACT	vi
TABLE DES MATIÈRES	viii
LISTE DES TABLEAUX	xi
LISTE DES FIGURES	xii
LISTE DES SIGLES ET ABRÉVIATIONS	xiv
LISTE DES ANNEXES	xv
CHAPITRE 1 INTRODUCTION	1
CHAPITRE 2 CADRE THÉORIQUE	4
2.1 La théorie cinétique des gaz	4
2.2 La fonction de distribution des particules f	7
2.2.1 Les moments de la fonction de distribution f	9
2.2.2 La distribution de Maxwell-Boltzmann	13
2.2.3 Les moments de la fonction de distribution à l'équilibre	13
2.3 L'équation de transport de Boltzmann	14
2.3.1 L'opérateur de collision BGK	15
2.3.2 Système d'équations intégral-différentielles	17
2.3.3 Dédution des équations de Navier-Stokes	18
CHAPITRE 3 DISCRÉTISATION NUMÉRIQUE ET IMPLÉMENTATION	19
3.1 La méthode de Boltzmann sur réseau	19
3.1.1 Les automates cellulaires	19
3.1.2 Discrétisation de la fonction f	20
3.1.3 Discrétisation des moments de f	22
3.1.4 Discrétisation de la fonction de distribution à l'équilibre	23
3.1.5 Calcul des poids d'intégration	24

3.1.6	L'équation de Boltzmann-BGK discrétisée	25
3.1.7	Approximation de la dérivée Lagrangienne de f_i	26
3.1.8	Vitesse du son	26
3.1.9	Calcul de la viscosité et déduction des équations de Navier-Stokes . .	27
3.1.10	La condition de stabilité de Courant–Friedrichs–Lewy	28
3.1.11	Système d'équations discrétisées	29
3.1.12	Conditions aux frontières et conditions initiales	29
3.1.13	Algorithme LBM	30
3.2	Implémentation	31
3.2.1	La librairie <i>open-source</i> Palabos	31
3.2.2	Les opérateurs de collision	32
3.2.3	Les modèles de conditions aux frontières	33
3.2.4	Paramètres de simulation d'écoulement incompressible	35
3.2.5	La loi de Darcy	37
3.2.6	Vérification de la méthode LB sur divers cas test	39
CHAPITRE 4	RÉSULTATS DE SIMULATIONS NUMÉRIQUES	40
4.1	Écoulement de Poiseuille 2D entre deux plaques	40
4.1.1	Effet de la longueur du domaine sur l'erreur numérique	42
4.1.2	Effet du modèle de conditions aux frontières sur l'erreur numérique .	44
4.1.3	Effet du nombre de Reynolds sur l'erreur numérique	44
4.1.4	Effet de la résolution sur le temps de calcul	47
4.2	Cavité carrée 2D avec paroi entraînée	49
4.2.1	Effet de la résolution sur la perte de masse	50
4.2.2	Effet du modèle de condition aux frontières sur la perte de masse . .	52
4.2.3	Effet de l'opérateur de collision sur la perte de masse	52
4.2.4	Effet de la résolution sur le champ de pression	52
4.3	Écoulement de Poiseuille 2D entre deux plaques avec obstacle cylindrique . .	57
4.3.1	Effet de la longueur du conduit sur le coefficient de traînée C_d	60
4.3.2	Effet de la résolution temporelle sur le coefficient de traînée C_d	60
4.3.3	Effet de la résolution spatiale sur le coefficient de traînée C_d	62
4.3.4	Comparaison des coefficients de traînée C_d obtenus	62
4.4	Écoulement de Poiseuille 3D à travers un cylindre	65
4.4.1	Analyse des sources d'erreurs sur la perméabilité calculée	67
4.4.2	Comparaison des types d'erreurs définis	71
4.5	Écoulement 3D à travers des réseaux de sphères	75

4.5.1	Effet de l'opérateur de collision et du temps de relaxation sur la perméabilité	78
4.5.2	Comparaison des perméabilités obtenues	80
CHAPITRE 5	CONCLUSION	82
RÉFÉRENCES	85
ANNEXES	89

LISTE DES TABLEAUX

Tableau 2.1	Système d'équations intégr-différentielles	17
Tableau 3.1	Système d'équations discrétisées	30
Tableau 3.2	Effet du paramètre contrôlant la résolution temporelle sur la dépendance à la résolution spatiale N_L du nombre de Mach	37
Tableau 4.1	Paramètres de simulation pour la section 4.2.4	53
Tableau 4.2	Paramètres de simulation pour les sections 4.3.1, 4.3.2 et 4.3.3 . . .	60
Tableau 4.3	Effet de la longueur du conduit l_x sur le coefficient de traînée C_d pour $\beta = 2$ et $N_L = 256$. $\Delta C_d(l_x/l_y) = C_d(l_x/l_y) - C_d(l_x/2l_y) $	60
Tableau 4.4	Effet de U_0 sur le coefficient de traînée C_d pour $\beta = 2$ et $N = 256$. $\Delta C_d(N_t) = C_d(N_t) - C_d(N_t/2) $	61
Tableau 4.5	Effet de la résolution spatiale N_L sur le coefficient de traînée C_d pour $\beta = 2$. $\Delta C_d(N_L) = C_d(N_L) - C_d(N_L/2) $	61
Tableau 4.6	Paramètres de simulation pour la section 4.3.4	62
Tableau 4.7	Comparaison des coefficients de traînée C_d pour différents β et Re . .	63
Tableau 4.8	Paramètres de simulation pour la section 4.4	65
Tableau 4.9	Largeur et porosité des réseaux cubiques utilisés afin de générer les géométries poreuses	75

LISTE DES FIGURES

Figure 3.1	Représentation des vecteurs de vitesse \mathbf{c}_i du schéma D2Q9 et de la distance δ_x entre les noeuds	22
Figure 4.1	Schéma du domaine de calcul pour l'écoulement de Poiseuille en 2D entre deux plaques	40
Figure 4.2	Erreur en fonction de l_x/l_y pour deux types de conditions aux frontières	43
Figure 4.3	Effet du modèle de conditions aux frontières sur l'erreur numérique $E_{\mathbf{u}}$	45
Figure 4.4	Effet du nombre de Reynolds sur l'erreur numérique $E_{\mathbf{u}}$	46
Figure 4.5	Erreur $E_{\mathbf{u}}$ en fonction du nombre de sites mis à jour, pour différents nombres de Reynolds et viscosités	48
Figure 4.6	Schéma du domaine de calcul pour la cavité 2D	50
Figure 4.7	Réduction de la perte de masse lors du raffinement de maillage	51
Figure 4.8	Effet du modèle de conditions aux frontières sur la perte de masse lors du raffinement du pas de temps	53
Figure 4.9	Effet de l'opérateur de collision sur la perte de masse lors du raffinement du pas de temps	54
Figure 4.10	Champs de pression adimensionnelle à $\tilde{t} = 200$ pour différents U_0 et nombres de Reynolds	56
Figure 4.11	Schéma du domaine de calcul pour l'écoulement de Poiseuille avec obstacle cylindrique	58
Figure 4.12	Section du domaine de simulation avec représentation d'un des maillages utilisés (différents diamètres menant aux mêmes maillages sont illustrés)	66
Figure 4.13	Schéma du domaine de calcul pour l'écoulement de Poiseuille en 3D à travers un cylindre	68
Figure 4.14	Erreur numérique maximale $E_{n,max}$ en fonction de la résolution pour le modèle BGK avec $\tau = 0.6$	72
Figure 4.15	Comparaison des différents types d'erreur en fonction de la résolution pour le modèle BGK avec $\tau = 0.6$	72
Figure 4.16	Erreur de maillage E_{maill} en fonction de la résolution pour différents opérateurs et temps de relaxation	73
Figure 4.17	Erreur de la méthode LB E_{LB} en fonction de la résolution pour différents opérateurs de collision et temps de relaxation	73
Figure 4.18	Géométrie des différents réseaux cubiques étudiés.	76

Figure 4.19	Effets de l'opérateur de collision et du temps de relaxation sur le nombre de Darcy lors du raffinement de maillage	79
Figure 4.20	Comparaison des coefficients de traînée K obtenus pour différents réseaux cubiques	81

LISTE DES SIGLES ET ABRÉVIATIONS

BGK	Bhatnagar-Gross-Krook
Da	Nombre de Darcy
LB	Lattice Boltzmann
LBM	Lattice Boltzmann Method
MRT	Multiple-Relaxation-Time
Re	Nombre de Reynolds

LISTE DES ANNEXES

ANNEXE A	MOMENTS DE LA FONCTION DE DISTRIBUTION À L'ÉQUILIBRE f^{eq}	89
ANNEXE B	DÉTAIL DU CODE D'UNE SIMULATION	94

CHAPITRE 1 INTRODUCTION

Le présent mémoire est consacré à la méthode de Boltzmann sur réseau ou *Lattice Boltzmann Method* (LBM) en anglais. Dans le présent document, cette méthode est souvent référée comme étant la méthode LB (Lattice-Boltzmann). Il s'agit d'une méthode basée sur l'équation de Boltzmann, une équation de transport de particules, permettant de simuler la dynamique d'un écoulement sur ordinateur. Un large éventail de déclinaisons de la méthode LB permet aussi de simuler différentes physiques d'écoulement, ou bien de permettre différents compromis entre précision, stabilité et temps de calcul. La méthode LB est une alternative prometteuse aux méthodes plus communes telles que les volumes finis [1] ou les éléments finis [2]. Elle tire son avantage du fait qu'elle décrit les écoulements au niveau particulaire, ce qui serait avantageux pour la simulation d'écoulements à physique complexe comme les écoulements à plusieurs phases [3] [4], le transport de particules [5], les écoulements en milieux poreux [6] [3] [7], etc.

Le but du travail accompli dans ce mémoire est tout d'abord de se familiariser avec la méthode LB, afin d'éventuellement viser une application concrète d'ampleur qui n'est pas complètement couverte dans le cadre de ce mémoire. Les cas tests abordés sont de complexité croissante afin de se familiariser progressivement avec l'implémentation informatique, les paramètres de simulations, les limites de la méthode, et l'analyse de résultats. Les cas tests choisis sont des cas tests classiques fréquemment abordés dans la littérature, ainsi l'auteur tente de faire preuve d'originalité sur les analyses qu'il en fait. Par exemple, le champ de pression et la conservation de la masse sont étudiés pour la cavité carrée plutôt que le champ de vitesse. D'autre part, les analyses effectuées sur les cas tests sont choisies afin de traiter de sujets variés. Ainsi on analyse des comportements numériques, dont la perte de masse, la représentation numérique des parois, la compressibilité de la méthode, et on aborde diverses variables physiques tels que les champs de vitesse, les champs de pression, les forces et la perméabilité.

Le présent mémoire élabore tout d'abord la théorie derrière la déclinaison la plus simple de la méthode de LB, soit celle permettant de résoudre les équations de Navier-Stokes pour un écoulement incompressible, instationnaire, isotherme et subsonique. La méthode est ensuite vérifiée sur quelques cas tests en terminant sur un cas en lien avec une application en hydrologie. L'application en hydrologie visée consiste à caractériser l'infiltration de l'eau dans les sols composés de granulats. Le sol étant considéré comme un milieu poreux, l'écoulement à travers celui-ci est simulé afin de calculer sa perméabilité. La perméabilité calculée pourrait

ensuite être intégrée à des modèles hydrologiques.

L'approche utilisée dans le présent document reflète le caractère multidisciplinaire de la méthode LB ; les disciplines concernées étant : la physique, l'analyse numérique, la programmation, et la mécanique des fluides numérique. La méthode LB trouve ensuite des applications dans diverses autres disciplines où il serait pertinent de simuler la dynamique d'un écoulement. Dans les prochains chapitres, ces disciplines sont abordées successivement à partir des modèles physiques jusqu'aux simulations.

Le Chapitre 2 aborde le cadre théorique de la méthode LB. Des modèles physiques permettant de décrire le comportement tant microscopique que macroscopique d'un écoulement sont présentés afin d'élaborer un système d'équations à résoudre.

Au Chapitre 3, l'analyse numérique nous permet de discrétiser ce système d'équations afin d'élaborer des solutions approximatives pouvant être calculées numériquement. Toujours au Chapitre 2, on aborde la formulation de l'algorithme permettant d'implémenter le calcul de ces approximations sous forme de programme informatique. Les simulations sont effectuées à l'aide de Palabos, une librairie *open-source* C++ optimisée pour le calcul en parallèle sur plusieurs processeurs. Le code du cinquième et dernier cas test portant sur l'écoulement à travers un réseau de sphères est détaillé dans l'annexe B afin de montrer un exemple utilisant la librairie Palabos.

Le Chapitre 4 relève pour sa part de la mécanique des fluides numériques, discipline qui englobe des composants de chacune des disciplines mentionnées ci-haut. Il s'agit en premier lieu de vérifier et valider les résultats de simulation découlant de la méthode LB. Ensuite, les limites de la méthode sont mises à l'épreuve afin de maîtriser le compromis entre précision, stabilité et temps de calcul. Finalement, on s'attarde à une de ses applications possibles, soit le calcul de la perméabilité de milieux poreux.

Au Chapitre 4, cinq cas tests sont abordés. Les cas tests ont été choisis en s'assurant de pouvoir comparer les résultats obtenus avec des solutions soient analytiques ou numériques. La complexité des cas tests abordés croît jusqu'au dernier cas test qui est en lien direct avec l'application de calcul de la perméabilité de milieux poreux granulaires. Les trois premiers cas tests sont des écoulements en 2 dimensions (2D) et visent à vérifier les capacités et les limites de la méthode LB. Les deux derniers cas tests sont des écoulements en 3 dimensions (3D) et s'attardent au calcul de la perméabilité de milieux poreux.

Le premier cas test consiste à simuler un écoulement de Poiseuille 2D entre 2 plaques planes. On s'attarde à l'effet de certains paramètres de simulation, tel que le type de condition à la frontière, sur les champs de vitesse et de pression. On observe entre autres les effets de la

compressibilité de la méthode.

Le second cas test consiste à simuler une cavité carrée 2D avec une paroi entraînée. On caractérise l'influence de certains des paramètres numériques, telles que les résolutions temporelle et spatiale, sur la non-conservation de la masse observée. L'effet du nombre de Reynolds et de la discrétisation spatiale sur le champ de pression est aussi considéré.

Le troisième cas test consiste à simuler un écoulement de Poiseuille 2D entre 2 plaques planes avec un obstacle cylindrique. La force de trainée exercée sur l'obstacle est calculée. Une analyse de sensibilité est effectuée sur les paramètres numériques afin de valider leur choix. Ces paramètres sont les résolutions temporelle et spatiale ainsi que la longueur du domaine de simulation. Les résultats de simulations sont ensuite comparés avec une référence de la littérature. Il est constaté que le maillage de type cartésien, à cellules identiques, semble être une contrainte importante.

Le quatrième cas test consiste à simuler un écoulement de Poiseuille 3D à travers un cylindre. On s'attarde à l'utilisation de la méthode afin de calculer la perméabilité d'un milieu poreux. Le cylindre représente un pore maillé à faible résolution. L'erreur sur la perméabilité calculée est évaluée, puis on tente d'en trouver la principale source. On constate que la principale source d'erreur est due à discrétisation de la géométrie en un maillage cartésien composé de cellules cubiques identiques.

Le cinquième cas test consiste à simuler un écoulement 3D à travers des réseaux compacts de sphères. On s'attarde encore une fois au calcul de la perméabilité de milieux poreux, mais cette fois-ci en vue d'une éventuelle application en hydrologie. Le but étant d'améliorer les modèles d'infiltration d'eau dans le sol en simulant l'écoulement au niveau microscopique. On considère alors un écoulement à travers différentes géométries idéalisées.

CHAPITRE 2 CADRE THÉORIQUE

La manière traditionnelle de décrire un écoulement en mécanique des fluides est par le biais de champs scalaires ou vectoriels de variables macroscopiques (masse volumique, vitesse, pression, température, etc.). Le fluide est considéré comme un milieu continu, même si en réalité il est composé de particules élémentaires et discrètes. Qu'arriverait-il si l'on décrivait un écoulement en assignant une équation du mouvement pour chacune des particules le constituant ? Obtiendrions-nous les mêmes prédictions que celles obtenues avec l'hypothèse de milieu continu ? La plupart des applications courantes en mécanique des fluides nécessitent de considérer au moins un nombre de particules de l'ordre du nombre d'Avogadro, c'est-à-dire $N_A \simeq 6 \times 10^{23}$. Un système d'équations d'une telle dimension est fort contraignant à résoudre. De plus, il n'est pas pertinent de connaître avec détail l'état microscopique d'un écoulement afin de décrire son état macroscopique. On pourrait cependant tirer avantage du très grand nombre de particules afin d'utiliser des outils statistiques pour traiter leurs nombreuses interactions. C'est ainsi que la mécanique statistique tente de faire un lien entre le monde microscopique des particules élémentaires et le monde macroscopique des milieux continus. La prochaine section présente la théorie cinétique des gaz ainsi que quelques-uns de ses principaux résultats. Il s'agit d'un modèle simple permettant à la fois de mettre en contexte l'équation de Boltzmann et d'établir un lien entre les paramètres d'un écoulement et les particules qui le composent.

2.1 La théorie cinétique des gaz

La théorie cinétique des gaz est une branche de la mécanique statistique qui concerne les gaz dans un régime particulier : suffisamment dilués au sens des forces d'attraction intermoléculaires, mais assez denses au sens du nombre de Knudsen. À partir d'un modèle d'interactions intermoléculaires, on parvient à expliquer certains comportements macroscopiques. Les quelques notions élémentaires de la théorie cinétique des gaz présentées dans la section qui suit sont basées sur le livre de Chapman et Cowling [8]. Voici un résumé des hypothèses nécessaires :

- Les molécules du gaz sont représentées par des particules sphériques rigides qui se déplacent et entrent en collisions élastiques les unes avec les autres. Ceci est une conséquence de l'hypothèse que le gaz est suffisamment dilué au sens des forces intermoléculaires. Les particules n'interagissent pas les unes avec les autres hormis lorsqu'elles entrent en collision. Les effets quantiques sont négligés et la dynamique des

molécules est traitée selon les lois de la mécanique classique : conservation de la masse, de la quantité de mouvement et de l'énergie.

- Le gaz est constitué d'un très grand nombre de molécules qui entrent très souvent en collision les unes avec les autres. Ceci est une conséquence de l'hypothèse que le gaz est suffisamment dense au sens du nombre de Knudsen. Lorsque le libre parcours moyen des molécules d'un gaz est suffisamment petit par rapport à une distance caractéristique de l'écoulement, le fluide peut être considéré comme un milieu continu. Les particules d'un gaz confiné dans un tel régime interagissent alors beaucoup plus souvent entre elles qu'avec les parois.
- Un chaos moléculaire subsiste dans le gaz. L'hypothèse de chaos moléculaire stipule que la vitesse et la position de deux particules de gaz qui entrent en collision ne sont pas corrélées. Cette hypothèse facilite grandement le traitement statistique des interactions intermoléculaires. La vitesse d'une particule est considérée comme une variable aléatoire indépendante de son historique de collisions.

On s'attarde à un gaz défini comme précédemment dont les particules sont sphériques. On considère également qu'elles ont toutes la même masse et le même diamètre. Un tel gaz est appelé un gaz idéal monoatomique.

On considère le volume V d'un gaz idéal confiné dont on suppose que l'état d'équilibre thermodynamique est atteint. L'énergie cinétique totale E_c des N particules contenues dans ce volume est définie ainsi :

$$E_c = \sum_{n=1}^N \frac{1}{2} m c_n^2 = \frac{1}{2} m \sum_{n=1}^N c_n^2 \quad (2.1)$$

Où m correspond à la masse des particules et c_n correspond à la vitesse de la particule n à un instant donné. Il est possible de définir la moyenne du carré des vitesses $\overline{c^2}$ ainsi :

$$\overline{c^2} = \frac{\sum_{n=1}^N c_n^2}{N} \quad (2.2)$$

En substituant l'équation (2.2) dans l'équation (2.1), on obtient alors :

$$E_c = \frac{1}{2} N m \overline{c^2} \quad (2.3)$$

Où E_c et N sont constants en fonction du temps puisque que le gaz est en équilibre, donc $\overline{c^2}$ est également constant. La pression d'un gaz confiné est déterminée en calculant la pression moyenne résultant de l'impact des particules sur les parois. Considérant le même gaz confiné, la seconde loi de Newton est appliquée aux particules qui impactent les parois. Il est alors

possible de déterminer la pression moyenne \bar{p} en fonction de la vitesse des particules c . [8]

$$\bar{p} = \frac{N m \overline{c^2}}{V 3} \quad (2.4)$$

Où m correspond à la masse des particules et N correspond au nombre de particules dans le volume confiné V . Pour un nombre de Knudsen suffisamment petit, l'hypothèse de milieu continu est introduite en définissant la masse volumique ainsi :

$$\rho = \frac{N m}{V} \quad (2.5)$$

Dans le cas limite d'un milieu continu, la pression moyenne \bar{p} devient la pression isostatique p . La pression isostatique correspond à la moyenne, sur les trois composantes spatiales, du flux de quantité de mouvement normal. L'équation (2.4) est reformulée ainsi :

$$p = \frac{\rho \overline{c^2}}{3} \quad (2.6)$$

La constante 3 apparaissant correspond aux 3 dimensions de l'espace, on obtiendrait un 2 pour un modèle en 2 dimensions. L'équation d'état des gaz parfaits est introduite afin de définir la température :

$$pV = Nk_B T \quad (2.7)$$

Où k_B est la constante de Boltzmann. En introduisant la masse volumique et la constante des gaz parfaits $R = k_B/m$, l'équation des gaz parfaits pour un milieu continu est réécrite ainsi :

$$p = \rho R T \quad (2.8)$$

À partir des équations (2.8) et (2.6) la température s'exprime en fonction de la vitesse des particules :

$$T = \frac{p}{\rho R} = \frac{\overline{c^2}}{3R} \quad (2.9)$$

L'énergie cinétique massique ϵ est calculée à partir de l'équation (2.1) :

$$\epsilon = \frac{E_c}{Nm} = \frac{1}{2} \overline{c^2} \quad (2.10)$$

Pour un gaz idéal monoatomique, l'énergie cinétique de translation est la seule forme d'énergie. Ceci signifie : aucune énergie potentielle et aucune énergie cinétique de rotation. L'énergie interne massique correspond donc à ϵ . Il est intéressant de constater que plusieurs variables macroscopiques (température, pression, énergie interne) sont tout simplement proportion-

nelles à la moyenne du carré de la vitesse des particules. Puisque le gaz est à l'équilibre thermodynamique, sa vitesse macroscopique et son énergie cinétique macroscopique sont nulles. Cependant, les particules possèdent tout de même une forme d'énergie cinétique chaotique au niveau microscopique.

Il est également intéressant de considérer les relations entre les paramètres microscopiques et les paramètres de diffusion macroscopiques qui interviennent dans les équations de Navier-Stokes. Il est possible de déterminer ces relations pour un gaz idéal qui n'est pas en équilibre thermodynamique. La diffusion de la quantité de mouvement est décrite en fonction de la viscosité dynamique définie selon la loi de comportement d'un fluide newtonien. La diffusion thermique est décrite en fonction de la conductivité thermique définie selon la loi de Fourier. [8]

Viscosité dynamique :

$$\mu = \frac{1}{3}\rho \bar{c} l \quad (2.11)$$

Conductivité thermique :

$$k = \frac{1}{3}\rho c_v \bar{c} l \quad (2.12)$$

Où c_v est la capacité thermique massique à volume constant et \bar{c} est la vitesse moyenne des particules en un point à un instant donné. Pour un gaz monoatomique, on retrouve à partir de (2.9) et (2.10) :

$$c_v = \frac{\epsilon}{T} = \frac{3R}{2} \quad (2.13)$$

Le paramètre l correspond au libre parcours moyen, c'est-à-dire la distance moyenne parcourue par une molécule entre deux collisions successives. Selon la théorie cinétique des gaz, le libre parcours moyen dépend de la concentration de particules, de leur diamètre et de leur vitesse moyenne [8]. Le temps de relaxation correspond au temps moyen entre deux collisions successives et celui-ci est étroitement lié au libre parcours moyen. Dans le contexte de la méthode LB, les paramètres de diffusions sont calibrés par rapport à une constante de temps également appelé temps de relaxation τ . Ce temps de relaxation, de manière similaire au libre parcours moyen, sert d'échelle de grandeur pour évaluer la fréquence des collisions.

2.2 La fonction de distribution des particules f

La mécanique Lagrangienne permet de définir les équations du mouvement de chacune des N particules d'un gaz. Le mouvement de chacune des particules est décrit par sa position $\mathbf{x} = [x, y, z]^T$ et sa vitesse $\mathbf{c} = [c_x, c_y, c_z]^T$ en fonction du temps. Chaque particule se déplace donc dans un espace de phase à 6 dimensions (\mathbf{x}, \mathbf{c}) en fonction du temps. En mécanique

statistique, plutôt que de décrire la vitesse et la position de chaque particule, on décrit le nombre de particules se trouvant à une position et à une vitesse donnée. La fonction de distribution des particules \tilde{f} décrit la concentration de particules pour un espace de phase à 6 dimensions (\mathbf{x}, \mathbf{c}) en fonction du temps. Elle est définie ainsi :

$$N = \iiint \iiint \tilde{f}(\mathbf{x}, \mathbf{c}, t) \, dx \, dy \, dz \, dc_x \, dc_y \, dc_z \quad (2.14)$$

Où N est le nombre de particules présentes dans un volume de l'espace de phase (\mathbf{x}, \mathbf{c}) au temps t . L'abréviation suivante est utilisée :

$$N = \iint \tilde{f}(\mathbf{x}, \mathbf{c}, t) \, d^3\mathbf{x} \, d^3\mathbf{c} \quad (2.15)$$

On emploie la notation $d^3\mathbf{x} = dx \, dy \, dz$ et $d^3\mathbf{c} = dc_x \, dc_y \, dc_z$ où $d^3\mathbf{x}$ et $d^3\mathbf{c}$ correspondent à des volumes infinitésimaux des espaces \mathbf{x} et \mathbf{c} respectivement. Puisqu'en mécanique des fluides on ne s'intéresse pas au nombre de particules, on élabore une version massique de l'équation (2.15). Pour des particules de même masse m , on retrouve la masse totale de particules m_{totale} d'un volume de l'espace de phase (\mathbf{x}, \mathbf{c}) ainsi :

$$m_{totale} = m \, N = m \iint \tilde{f}(\mathbf{x}, \mathbf{c}, t) \, d^3\mathbf{x} \, d^3\mathbf{c} \quad (2.16)$$

On assimile ensuite la masse des particules m à la fonction de distribution f :

$$f := m\tilde{f} \quad (2.17)$$

Par exemple, dans le cas où l'on s'intéresse à la masse d'un volume V de l'espace des positions \mathbf{x} , l'équation (2.16) est reformulée ainsi :

$$m_{totale} = N \, m = \int_{\mathbf{c} \in \mathbb{R}^3} \int_{\mathbf{x} \in V} f(\mathbf{x}, \mathbf{c}, t) \, d^3\mathbf{x} \, d^3\mathbf{c} \quad (2.18)$$

Où le domaine d'intégration des vitesses microscopiques est l'ensemble des réels pour les 3 composantes de \mathbb{R}^3 puisque la masse de chaque particule est considérée, indépendamment de sa vitesse. Les unités de f sont donc : $\frac{\text{kg}}{\text{m}^3 \cdot (\text{m}^3/\text{s}^3)}$. En adoptant l'hypothèse de milieu continu, la fonction de distribution f devient une fonction continue de manière similaire à une fonction de densité de probabilité. La fonction f est en quelque sorte une description du flux massique des particules dans l'espace. Afin de considérer des particules de masses différentes, il suffit d'attribuer une fonction de distribution différente pour chaque masse différente.

2.2.1 Les moments de la fonction de distribution f

Les variables macroscopiques sont calculées à partir des moments de la fonction de distribution f . Les relations entre les variables microscopiques et macroscopiques sont similaires à celles de la théorie cinétique des gaz et dépendent principalement de la distribution des vitesses des particules. Les variables macroscopiques en mécanique des fluides sont typiquement évaluées par rapport à la vitesse locale \mathbf{u} de l'écoulement (repère Eulérien) alors que la vitesse des particules \mathbf{c} est évaluée par rapport à un même repère inertiel pour tout l'écoulement (repère Lagrangien). Hormis le cas de la vitesse macroscopique \mathbf{u} , les paramètres macroscopiques évalués par rapport au repère Lagrangien sont identifiés par l'indice $_0$. Bien que le domaine d'intégration soit omis pour l'expression des moments, celui-ci correspond toujours à l'ensemble des réels \mathbb{R}^D où D correspond au nombre de dimensions de l'espace des positions. Le cas particulier à $D = 3$ dimensions est considéré pour l'expression des moments qui suivent.

La masse volumique ρ correspond à la concentration massique de toutes les particules en un point :

$$\rho(\mathbf{x}, t) = \int f d^3\mathbf{c} \quad (2.19)$$

La quantité de mouvement macroscopique $\rho\mathbf{u}$ correspond à la somme vectorielle de la quantité de mouvement de toutes les particules :

$$\rho\mathbf{u}(\mathbf{x}, t) = \int f\mathbf{c} d^3\mathbf{c} \quad (2.20)$$

La densité d'énergie cinétique macroscopique ϵ est obtenue en sommant l'énergie cinétique de chaque particule :

$$\rho\epsilon_0(\mathbf{x}, t) = \int \frac{1}{2}f|\mathbf{c}|^2 d^3\mathbf{c} \quad (2.21)$$

$$\rho\epsilon(\mathbf{x}, t) = \int \frac{1}{2}f|\mathbf{c} - \mathbf{u}|^2 d^3\mathbf{c} \quad (2.22)$$

Où $|\mathbf{c}|$ correspond à la norme euclidienne du vecteur vitesse \mathbf{c} , soit $|\mathbf{c}|^2 = \mathbf{c} \cdot \mathbf{c} = \mathbf{c}^\top \mathbf{c}$. Par substitution on retrouve :

$$\epsilon_0 = \epsilon + \frac{|\mathbf{u}|^2}{2} \quad (2.23)$$

Le flux de quantité de mouvement permet de calculer le tenseur de contraintes internes $\boldsymbol{\sigma}$:

$$\boldsymbol{\sigma}_0(\mathbf{x}, t) = - \int f\mathbf{c} \otimes \mathbf{c} d^3\mathbf{c} \quad (2.24)$$

$$\boldsymbol{\sigma}(\mathbf{x}, t) = - \int f(\mathbf{c} - \mathbf{u}) \otimes (\mathbf{c} - \mathbf{u}) d^3\mathbf{c} \quad (2.25)$$

Où \otimes est le produit externe où $\mathbf{u} \otimes \mathbf{v} = \mathbf{u}\mathbf{v}^\top$. En notation indicielle :

$$\sigma_{ij}(\mathbf{x}, t) = - \int f(c_i - u_i)(c_j - u_j) d^3\mathbf{c} \quad (2.26)$$

$$\sigma_{0ij}(\mathbf{x}, t) = - \int f c_i c_j d^3\mathbf{c} \quad (2.27)$$

Le signe négatif apparait puisque le tenseur de contrainte est défini en évaluant la force exercée sur l'élément de fluide. On observe par commutativité que le tenseur $\boldsymbol{\sigma}$ est symétrique. Par substitution, on obtient :

$$\sigma_{0ij} = \sigma_{ij} - \rho u_i u_j \quad (2.28)$$

La pression isostatique est définie comme étant le flux de quantité de mouvement normal moyen et peut être exprimée ainsi :

$$p(\mathbf{x}, t) = - \frac{(\sigma_{xx} + \sigma_{yy} + \sigma_{zz})}{3} \quad (2.29)$$

Il est possible de lier la pression à l'énergie cinétique :

$$p = -\frac{1}{3}(\sigma_{xx} + \sigma_{yy} + \sigma_{zz}) = \frac{1}{3} \int f (\mathbf{c} - \mathbf{u}) \cdot (\mathbf{c} - \mathbf{u}) d\mathbf{c} = \frac{2}{3} \rho \epsilon \quad (2.30)$$

Le tenseur de contraintes $\boldsymbol{\sigma}$ peut être décomposé ainsi :

$$\boldsymbol{\sigma} = \boldsymbol{\tau} - p\mathbf{I} \quad (2.31)$$

Où $\boldsymbol{\tau}$ correspond au tenseur de contraintes visqueuses. Finalement, on évalue le flux d'énergie cinétique :

$$\mathbf{q}_0 = \int \frac{1}{2} f |\mathbf{c}|^2 \mathbf{c} d^3\mathbf{c} \quad (2.32)$$

$$\mathbf{q} = \int \frac{1}{2} f |\mathbf{c}|^2 (\mathbf{c} - \mathbf{u}) d^3\mathbf{c} \quad (2.33)$$

Par substitution, on obtient :

$$\mathbf{q}_0 = \mathbf{q} - \boldsymbol{\sigma}\mathbf{u} + \mathbf{u} \left(\rho \epsilon + \frac{\rho |\mathbf{u}|^2}{2} \right) \quad (2.34)$$

Pour un gaz idéal monoatomique, la totalité de l'énergie est emmagasinée sous forme d'énergie cinétique de translation, donc ϵ correspond à l'énergie interne et \mathbf{q} correspond au flux de chaleur.

À partir des flux de : masse $\rho\mathbf{u}$, quantité de mouvement $-\boldsymbol{\sigma}_0$ et énergie \mathbf{q}_0 , il est possible de

formuler les équations de transport de ces propriétés. Soit la forme différentielle de l'équation de transport :

$$\frac{\partial \phi}{\partial t} + \nabla \cdot \mathbf{j} = s \quad (2.35)$$

Où $\phi = \phi(\mathbf{x}, t)$ est une propriété intensive transportée, $\mathbf{j} = [j_x(\mathbf{x}, t), j_y(\mathbf{x}, t), j_z(\mathbf{x}, t)]^\top$ est le flux de cette propriété par rapport au repère Lagrangien, et $s = s(\mathbf{x}, t)$ est un terme source. Une propriété intensive est indépendante de la taille de l'élément de fluide considéré, ce qui permet de traiter un élément de taille infinitésimale. Le produit scalaire de l'opérateur Nabla est défini ainsi : $\nabla \cdot \mathbf{j} = \frac{\partial j_x}{\partial x} + \frac{\partial j_y}{\partial y} + \frac{\partial j_z}{\partial z}$.

À partir de $\phi = \rho$ et $\mathbf{j} = \rho \mathbf{u}$, on retrouve l'équation de transport de la masse :

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0 \quad (2.36)$$

À partir de $\phi = \rho \mathbf{u}$ et $\mathbf{j} = \boldsymbol{\sigma}_0$ on retrouve l'équation de transport de la quantité de mouvement :

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot (-\boldsymbol{\tau} + p\mathbf{I} + \rho \mathbf{u} \otimes \mathbf{u}) = \mathbf{f} \quad (2.37)$$

À partir de $\phi = \rho \epsilon_0$ et $\mathbf{j} = \mathbf{q}_0$ on retrouve l'équation de transport de l'énergie :

$$\frac{\partial \rho \epsilon_0}{\partial t} + \nabla \cdot (\mathbf{q} + (-\boldsymbol{\tau} + p\mathbf{I}) \mathbf{u} + \mathbf{u} \rho \epsilon_0) = Q_s \quad (2.38)$$

Où \mathbf{f} et Q_s sont respectivement les termes source de quantité de mouvement et d'énergie. Une forme plus courante des équations de transport est celle pour laquelle les termes diffusifs sont assimilés au membre de droite :

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0 \quad (2.39)$$

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) = -\nabla p + \nabla \cdot \boldsymbol{\tau} + \mathbf{f} \quad (2.40)$$

$$\frac{\partial \rho \epsilon_0}{\partial t} + \nabla \cdot (\rho \epsilon_0 \mathbf{u}) = -\nabla \cdot (\mathbf{u} p) - \nabla \cdot \mathbf{q} + \nabla \cdot (\boldsymbol{\tau} \mathbf{u}) + Q_s \quad (2.41)$$

L'opérateur Nabla est défini ainsi : $\nabla = \left[\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z} \right]$. On remarque donc qu'il est possible d'exprimer les trois équations de transport de la mécanique des fluides à partir des moments de la fonction de distribution f . Les moments de la fonction de distribution peuvent être calculés à partir des équations (2.42) et (2.43) :

Par rapport au repère Lagrangien :

$$\mathbf{M}_n(\mathbf{x}, t) = \int_{\mathbb{R}^D} \mathbf{c}^{\otimes n} f(\mathbf{x}, \mathbf{c}, t) d^D \mathbf{c} \quad (2.42)$$

Par rapport au repère Eulérien (moments centrés) :

$$\mathbf{m}_n(\mathbf{x}, t) = \int_{\mathbb{R}^D} (\mathbf{c} - \mathbf{u})^{\otimes n} f(\mathbf{x}, \mathbf{c}, t) d^D \mathbf{c} \quad (2.43)$$

Où \otimes^n correspond à la puissance tensorielle, c'est-à-dire une puissance dont le produit est le produit externe \otimes . L'opérateur $d^D \mathbf{c}$ est la généralisation de $d^3 \mathbf{c}$, c'est-à-dire que $d^2 \mathbf{c} = dc_x dc_y$. Les moments de f sont, de manière générale, des combinaisons linéaires des différentes composantes des termes de \mathbf{M}_n ou \mathbf{m}_n et où n correspond à l'ordre des moments qui en découlent.

Les moments sont une manière réductrice de décrire la dispersion des particules à travers l'espace des vitesses \mathbf{c} et permettent de faire un lien avec les variables macroscopiques qui évoluent dans l'espace de phase à 3 dimensions \mathbf{x} .

Seuls 13 moments d'ordre 0 à 3 sont nécessaires afin d'évaluer, en fonction de f , les termes des 3 équations de transport (2.39) (2.40) (2.41) :

$$\rho, u, v, w, \sigma_{xx}, \sigma_{yy}, \sigma_{zz}, \sigma_{xy}, \sigma_{yz}, \sigma_{xz}, q_x, q_y, q_z \quad (2.44)$$

Ces moments sont : ρ , les trois composantes de \mathbf{u} , 6 des 9 composantes de $\boldsymbol{\sigma}$ étant donné la symétrie de ce dernier, et les trois composantes de \mathbf{q} . Les variables ϵ_0 , p et $\boldsymbol{\tau}$ s'expriment en fonction de $\boldsymbol{\sigma}$ à partir des équations (2.23), (2.30) et (2.31).

L'équation de transport de la masse ne fait qu'intervenir des moments d'ordre 0 et 1. L'équation de conservation de la quantité de mouvement ne fait qu'intervenir des moments d'ordre 0, 1 et 2. L'équation de transport de l'énergie ne fait qu'intervenir des moments d'ordre 0, 1, 2 et 3.

Ceci donne un avant-goût quelque peu heuristique par rapport aux « degrés de liberté » (moments) de la fonction f qui ne sont pas pertinents dans la représentation des équations de transport de la mécanique des fluides. Lors de la discrétisation numérique de la fonction f dans le contexte de la méthode LB, ces « degrés de liberté » peuvent être soit négligés ou soit employés comme paramètres numériques afin de contrôler la précision et la stabilité de la méthode [9].

2.2.2 La distribution de Maxwell-Boltzmann

La distribution de Maxwell-Boltzmann est une fonction de densité de probabilité permettant de décrire la distribution de vitesse des particules pour un gaz idéal en équilibre thermodynamique. Cette distribution ne dépend ni de la position ni du temps. Elle découle de la supposition que chaque composante de la vitesse d'une particule est une variable aléatoire suivant une loi normale. Elle permet d'exprimer l'état à l'équilibre de la fonction f :

$$f^{eq}(\mathbf{c}) = \frac{\rho}{(2\pi RT)^{D/2}} \exp \left[-\frac{|\mathbf{c} - \mathbf{u}|^2}{2RT} \right] \quad (2.45)$$

Où D correspond au nombre de dimensions de l'espace. La fonction de distribution à l'équilibre $f^{eq}(\mathbf{c})$ suit donc une loi de chi-carré à D degrés de liberté.

2.2.3 Les moments de la fonction de distribution à l'équilibre

La fonction de distribution à l'équilibre fait intervenir 3 paramètres variables, soit ρ , u et T puisque R est une constante pour un gaz donné. Les liens entre les paramètres de cette distribution et les paramètres macroscopiques sont déterminés en calculant les moments de la fonction de distribution à l'équilibre dénoté par l'indice eq . Les détails du calcul de la masse volumique, de la vitesse macroscopique et de l'énergie cinétique massique de f^{eq} se retrouvent dans l'annexe A.

La masse volumique ρ apparaissant dans l'équation (2.45) correspond bien à la masse volumique calculée à partir de l'équation (2.19) :

$$\rho_{eq}(\mathbf{x}, t) = \int f^{eq} d^3\mathbf{c} = \rho \quad (2.46)$$

$$\rho_{eq} = \rho \quad (2.47)$$

La vitesse \mathbf{u} apparaissant dans l'équation (2.45) correspond bien à la vitesse macroscopique calculée à partir de l'équation (2.20) :

$$\rho_{eq}\mathbf{u}_{eq}(\mathbf{x}, t) = \int f^{eq}\mathbf{c} d^3\mathbf{c} = \rho\mathbf{u} \quad (2.48)$$

$$\mathbf{u}_{eq} = \mathbf{u} \quad (2.49)$$

La température T apparaissant dans l'équation (2.45) s'exprime en fonction de l'énergie cinétique à partir de l'équation (2.22) :

$$\rho_{eq}\epsilon_{eq}(\mathbf{x}, t) = \frac{1}{2} \int f^{eq} |\mathbf{c} - \mathbf{u}|^2 d^3\mathbf{c} = \rho \frac{3}{2} RT \quad (2.50)$$

$$\epsilon_{eq} = \frac{3}{2}RT \quad (2.51)$$

La distribution (2.45) est donc définie de manière unique en imposant 3 moments d'ordre 0, 1 et 2. À partir de l'équation (2.51) et de l'équation (2.30), on retrouve que la pression de la fonction de distribution à l'équilibre respecte l'équation d'état des gaz parfaits :

$$p_{eq} = \frac{2}{3}\rho_{eq}\epsilon_{eq} = \rho RT \quad (2.52)$$

2.3 L'équation de transport de Boltzmann

Sous sa forme la plus minimaliste, l'équation de Boltzmann est définie ainsi :

$$\frac{df}{dt} = C(f) \quad (2.53)$$

C'est-à-dire que la dérivée totale de f est égale à l'opérateur de collision $C(f)$ avec $f = f(\mathbf{x}, \mathbf{c}, t)$. L'opérateur de collision représente l'effet des collisions entre les particules sur la fonction de distribution f . On développe la dérivée de f en suivant la règle de dérivation en chaîne :

$$\frac{df}{dt} = \frac{\partial f}{\partial t} + \frac{d\mathbf{x}}{dt} \cdot \nabla_{\mathbf{x}} f + \frac{d\mathbf{c}}{dt} \cdot \nabla_{\mathbf{c}} f \quad (2.54)$$

Où $\nabla_{\mathbf{x}} f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z} \right]$ et $\nabla_{\mathbf{c}} f = \left[\frac{\partial f}{\partial c_x}, \frac{\partial f}{\partial c_y}, \frac{\partial f}{\partial c_z} \right]$. On effectue ensuite la substitution suivante :

$$\frac{d\mathbf{x}}{dt} = \mathbf{c} \quad (2.55)$$

On introduit la seconde loi de Newton, où \mathbf{F} est la force externe exercée sur les particules et \mathbf{g} est l'accélération qui en découle :

$$\mathbf{F} = m \frac{d\mathbf{c}}{dt} = m\mathbf{g} \quad (2.56)$$

On obtient alors une formulation moins compacte et plus courante de l'équation de transport de Boltzmann :

$$\frac{\partial f}{\partial t} + \mathbf{c} \cdot \nabla_{\mathbf{x}} f + \mathbf{g} \cdot \nabla_{\mathbf{c}} f = C(f) \quad (2.57)$$

L'équation de Boltzmann est donc une équation de transport de f dont la vitesse advective est la vitesse particulaire \mathbf{c} (repère Lagrangien) plutôt que la vitesse macroscopique \mathbf{u} (repère Eulérien). Il est possible de déduire les 3 équations de transport de la mécanique des fluides (2.39), (2.40), (2.41) à partir des moments de l'équation de Boltzmann [10].

2.3.1 L'opérateur de collision BGK

Il existe plusieurs développements permettant de calculer l'effet des collisions représenté par $C(f)$ sur la distribution des particules f [11]. Pour les méthodes LB, ce sont les opérateurs à temps de relaxation qui sont le plus couramment employés. Le plus simple est l'opérateur Bhatnagar-Gross-Krook (BGK) à temps de relaxation unique. L'opérateur de collision est linéarisé en introduisant la fonction de distribution à l'équilibre ainsi qu'un temps de relaxation τ .

$$C(f) = \frac{1}{\tau}(f^{eq} - f) \quad (2.58)$$

Le critère d'admissibilité de cette approximation est que l'échelle de temps de diffusion soit beaucoup plus petite que l'échelle de temps d'advection [12] ; bien que les particules voyagent à de très grandes vitesses, elles parcourent une très petite distance avant d'entrer en collision entre elles. Autrement dit, le libre parcours moyen doit être beaucoup plus petit qu'une longueur caractéristique de l'écoulement. De manière similaire à l'hypothèse de milieu continu, le nombre de Knudsen doit donc être suffisamment petit. Les nombreuses collisions ont pour effet de ramener très rapidement la fonction f à un état d'équilibre local f^{eq} . Le gaz est donc considéré comme étant à l'équilibre localement même s'il ne l'est pas globalement. Le temps de relaxation τ est un paramètre permettant de calibrer la vitesse à laquelle cet équilibre local est atteint. On obtient l'équation de Boltzmann-BGK (2.59) en intégrant l'approximation BGK (2.58) à l'équation de Boltzmann :

$$\frac{\partial f}{\partial t} + \mathbf{c} \cdot \nabla_{\mathbf{x}} f + \mathbf{g} \cdot \nabla_{\mathbf{c}} f = \frac{1}{\tau}(f^{eq} - f) \quad (2.59)$$

On observe la présence des quatre principaux mécanismes de transport. Pour le membre de gauche : le premier terme correspond à l'accumulation, le second correspond à l'advection et le troisième correspond à un terme source. Pour le membre de droite, l'opérateur de collision correspond à un terme de diffusion. Pour un gaz idéal, les mêmes lois physiques régissent les comportements macroscopiques et microscopiques. Ceci implique que la masse, la quantité de mouvement et l'énergie sont conservés lors des collisions. Puisque l'opérateur BGK est introduit artificiellement dans l'équation de transport, on s'assure qu'il respecte les lois de conservations. On impose alors les lois de conservation à l'opérateur BGK :

Conservation de la masse :

$$\int \frac{1}{\tau}(f^{eq} - f) d^3\mathbf{c} = 0 \Rightarrow \int f^{eq} d^3\mathbf{c} = \int f d^3\mathbf{c} \quad (2.60)$$

En substituant l'équation (2.19) dans l'équation précédente, on obtient :

$$\rho_{eq} = \rho = \int f d^3\mathbf{c} \quad (2.61)$$

Conservation de la quantité de mouvement :

$$\int \frac{1}{\tau} (f^{eq} - f) \mathbf{c} d^3\mathbf{c} = \mathbf{0} \Rightarrow \int f^{eq} \mathbf{c} d^3\mathbf{c} = \int f \mathbf{c} d^3\mathbf{c} \quad (2.62)$$

En substituant l'équation (2.20) dans l'équation précédente, on obtient :

$$\rho_{eq} \mathbf{u}_{eq} = \rho \mathbf{u} = \int f \mathbf{c} d^3\mathbf{c} \quad (2.63)$$

Conservation de l'énergie :

$$\int \frac{1}{\tau} (f^{eq} - f) |\mathbf{c}|^2 d^3\mathbf{c} = 0 \Rightarrow \int f^{eq} |\mathbf{c}|^2 d^3\mathbf{c} = \int f |\mathbf{c}|^2 d^3\mathbf{c} \quad (2.64)$$

En substituant les équations (2.21) et (2.23) dans l'équation précédente, on obtient :

$$\rho_{eq} \epsilon_{eq} + \rho_{eq} \frac{|\mathbf{u}_{eq}|^2}{2} = \rho \epsilon + \rho \frac{|\mathbf{u}|^2}{2} \quad (2.65)$$

En substituant (2.51) dans l'équation (2.65) avec $D = 3$, on obtient :

$$\rho_{eq} \epsilon_{eq} = \rho \frac{3}{2} R T = \rho \epsilon = \frac{1}{2} \int f |\mathbf{c} - \mathbf{u}|^2 d^3\mathbf{c} \quad (2.66)$$

Les équations (2.61), (2.63) et (2.65) permettent, à partir de f , de définir de manière unique les paramètres apparaissant dans la fonction f^{eq} de l'opérateur BGK et de s'assurer que celui-ci respecte les lois de conservation.

Il est possible de retrouver l'équation d'état des gaz parfaits à partir de (2.52) et (2.30) en ajoutant la contrainte de conservation de l'énergie (2.66) appliquée à l'opérateur BGK :

$$p_{eq} = \frac{2}{3} \rho_{eq} \epsilon_{eq} = \frac{2}{3} \rho \epsilon = p = \rho R T \quad (2.67)$$

On conclut alors que l'opérateur BGK contraint la fonction f à suivre l'équation d'état des gaz parfaits.

Tableau 2.1 Système d'équations intégrro-différentielles

Propriété transportée :	$f = f(\mathbf{x}, \mathbf{c}, t)$
Équation de transport :	$\frac{\partial f}{\partial t} + \mathbf{u} \cdot \nabla_{\mathbf{x}} f + \mathbf{g} \cdot \nabla_{\mathbf{c}} f = C(f)$
Condition initiale :	$f(\mathbf{x}, \mathbf{c}, t = 0) = f_0$
Loi de comportement :	$C(f) = \frac{1}{\tau}(f^{eq} - f)$ $f^{eq} = \frac{\rho}{(2\pi RT)^{D/2}} \exp\left[\frac{- \mathbf{c} - \mathbf{u} ^2}{2RT}\right]$
Lois de conservation :	$\rho(\mathbf{x}, t) = \int f d^3\mathbf{c}$ $\rho\mathbf{u}(\mathbf{x}, t) = \int f\mathbf{c} d^3\mathbf{c}$ $\rho\frac{D}{2}RT(\mathbf{x}, t) = \frac{1}{2} \int f \mathbf{c} - \mathbf{u} ^2 d^3\mathbf{c}$

2.3.2 Système d'équations intégrro-différentielles

Le tableau 2.1 présente un récapitulatif du système d'équations permettant de résoudre l'équation de Boltzmann-BGK. Il s'agit d'un problème de Cauchy : on désire prévoir l'évolution du champ f en fonction du temps grâce à une équation aux dérivées partielles et une condition initiale. Le traitement des conditions aux frontières n'est pas abordé avec détail ; il s'agit de entre autres de définir un opérateur de collision $C(f)$ local reflétant l'effet des collisions des particules sur des parois solides.

Bien que l'équation de Boltzmann puisse s'appliquer au transport d'une très grande variété de particules, la distribution de Maxwell-Boltzmann introduite est basée sur l'hypothèse de gaz idéal et permet de retrouver l'équation d'état des gaz parfaits. D'autre part, il n'est pas simplificateur de développer une version incompressible ou stationnaire de l'équation de Boltzmann. Imposer l'incompressibilité nécessite de fixer ρ , ce qui correspond à imposer une contrainte sur un moment de la propriété transportée f . De même, pour considérer un écoulement stationnaire, il ne suffit pas de poser $\frac{\delta f}{\delta t} = 0$ mais plutôt d'imposer que la vitesse macroscopique \mathbf{u} soit constante localement. Ceci nécessite d'imposer un moment d'ordre 1 de la propriété transportée f . On alourdit mathématiquement la résolution en ajoutant ces contraintes, il est donc plus simple de conserver une forme instationnaire et compressible de l'équation. Il suffit de vérifier que l'on travaille dans un régime incompressible à faible nombre de Mach et que la solution obtenue ne possède aucun comportement instationnaire.

2.3.3 Dédution des équations de Navier-Stokes

L'expansion de Chapman-Enskog permet de retrouver les équations de Navier-Stokes à partir de l'équation de Boltzmann. Cette méthode consiste à proposer une solution de f sous la forme d'une série de puissances. Des expansions d'ordre 0, 1 et 2 permettent de retrouver respectivement : les équations d'Euler, les équations de Navier-Stokes et les équations de Burnett. [13] Une expansion d'ordre supérieur permet de retrouver les équations de super-Burnett.

CHAPITRE 3 DISCRÉTISATION NUMÉRIQUE ET IMPLÉMENTATION

3.1 La méthode de Boltzmann sur réseau

Ce chapitre aborde les différentes méthodes numériques employées par la méthode LB afin de discrétiser l'équation de Boltzmann-BGK. Il s'agit entre autres de : discrétiser les espaces continus en espaces discrets, approximer les dérivées par des différences finies, évaluer les intégrales à partir de quadratures et approximer les fonctions par des polynômes. La structure de l'algorithme LBM est basée sur un schéma numérique explicite ainsi qu'une séparation entre l'advection et la diffusion (collision) des particules. D'autre part, la méthode LB est validée en s'assurant qu'elle résout les équations de Navier-Stokes, plutôt que l'équation de Boltzmann. En effet, on cherche une solution aux équations de Navier-Stokes exprimée en terme du champ de distribution de particules f plutôt qu'en terme de champs de vitesse, pression et température. Dans le présent chapitre, à partir de l'équation de Boltzmann-BGK, on déduit l'algorithme de résolution pour le cas d'un écoulement isotherme et incompressible. Les automates cellulaires sont tout d'abord brièvement présentés afin de décrire, d'un point de vue historique, l'origine du schéma numérique de discrétisation de la méthode LB.

3.1.1 Les automates cellulaires

Les automates cellulaires sont considérés comme étant à l'origine des gaz sur réseau qui ont à leur tour conduit au développement de la méthode LB [14]. Dans le cas d'un automate cellulaire, il s'agit de considérer un réseau de cellules ; à chaque cellule du réseau est attribué un état. Le réseau évolue ensuite par incrément que l'on appelle pas de temps. Une série de règles permettent de déterminer l'état des cellules au pas de temps suivant, à partir de l'état des cellules au pas de temps actuel. Ce qui est intéressant à observer est l'apparition de comportements macroscopiques sur des groupes de cellules [15]. Ces comportements sont difficilement prévisibles en observant les règles élémentaires de mise à jour des cellules. On conclut alors que de simples règles de mise à jour locales provoquent des comportements globaux complexes à travers le réseau de cellules.

Les gaz sur réseau

La méthode des gaz sur réseau est un type d'automate cellulaire qui a pour but de représenter le comportement d'un gaz idéal. Le réseau de cellules correspond à une discrétisation de l'espace. Chaque cellule peut être occupée par une ou plusieurs particules qui voyagent à une

certaine vitesse donnée. Le nombre de particules par cellule ainsi que leur vitesse respective correspondent aux différents états admissibles des cellules. Les règles de mises à jour sont des règles d'advection et de collision (diffusion) des particules. Il y a donc deux étapes de mises à jour qui se succèdent alternativement. Pendant l'étape d'advection, les particules sont advectées dans les cellules voisines en fonction de la direction de leur vitesse. Pendant l'étape de collision, les particules entrent en collision les unes avec les autres, ce qui change la direction de leur vitesse. Durant ces deux étapes, on s'assure que les lois de conservation sont respectées. En choisissant attentivement les règles de mise à jour ainsi que la structure du réseau, il est possible de récupérer les équations de Navier-Stokes dans le cas limite macroscopique [16]. On conclut alors, de manière similaire aux automates cellulaires, que de simples règles de mises à jour locales (advection et collision) permettent d'engendrer un comportement global complexe (équations de Navier-Stokes).

Passage des automates cellulaires à la méthode de Boltzmann sur réseau

La méthode LB est née de l'adaptation de la méthode des gaz sur réseau pour la fonction de distribution de particules f [17]. Plutôt que de travailler avec des particules discrètes, on travaille avec des flux de particules décrits par f . L'algorithme de résolution correspond encore une fois à une alternance entre deux étapes, advection et collision, qui respectent les lois de conservation. De manière similaire aux gaz sur réseau, la méthode est vérifiée en s'assurant qu'on récupère les équations de Navier-Stokes dans le cas limite.

3.1.2 Discrétisation de la fonction f

Chaque dimension de la fonction de distribution $f(\mathbf{x}, \mathbf{c}, t)$ est discrétisée. L'espace \mathbf{x} est discrétisé sur un maillage cartésien structuré à pas d'espace δ_x constant. Le temps est également discrétisé et l'intervalle δ_t correspond au temps écoulé entre deux pas de temps successifs. L'espace des vitesses est discrétisé de sorte que les particules ne possèdent qu'une quantité finie de vitesses différentes admissibles. L'espace \mathbf{c} devient \mathbf{c}_i , où chaque i correspond à une différente vitesse admissible.

L'algorithme LBM est normalement implémenté avec des pas de temps δ_t et d'espace δ_x unitaires. Pour un problème donné, on aura alors trois échelles d'unités : physiques, adimensionnelles et LBM. Les unités LBM sont des unités sans dimensions au même titre que les unités adimensionnelles, cependant les grandeurs caractéristiques des unités LBM sont de nature numérique et dépendent de la finesse de discrétisation.

Schémas DnQm

Les schémas DnQm correspondent aux différentes manières de discrétiser l'espace des vitesses de la fonction de distribution f où « n » correspond au nombre de dimensions de l'espace des positions et « m » correspond au nombre de vitesses différentes de l'espace des vitesses discrétisé. Le schéma D2Q9 correspond donc à un schéma à 2 dimensions et 9 vitesses.

Étant donnée la nature du schéma numérique, une contrainte s'applique sur la discrétisation des vitesses : les particules sont advectées d'une cellule à une cellule voisine entre chaque pas de temps. Cette contrainte est définie ainsi :

$$\boldsymbol{\delta}_{\mathbf{x},i} = \mathbf{c}_i \delta_t \quad (3.1)$$

Où $\boldsymbol{\delta}_{\mathbf{x},i}$ correspond au déplacement effectué par les particules possédants une vitesse \mathbf{c}_i durant un pas de temps d'une durée δ_t . En utilisant les unités LBM, soit $\delta_t = 1$, l'équation (3.1) se réécrit ainsi :

$$\boldsymbol{\delta}_{\mathbf{x},i} = \mathbf{c}_i \quad (3.2)$$

Les vecteurs vitesse \mathbf{c}_i correspondent donc aux vecteurs déplacement $\boldsymbol{\delta}_{\mathbf{x},i}$ des particules entre chaque pas de temps. Ceux-ci sont représentés pour le schéma D2Q9 à la figure 3.1. Les valeurs des vitesses \mathbf{c}_i du schéma D2Q9 sont les suivantes :

$$\begin{array}{lll} \mathbf{c}_0 = [0, 0]^\top & \mathbf{c}_1 = -\mathbf{c}_3 = [c, 0]^\top & \mathbf{c}_5 = -\mathbf{c}_7 = [c, c]^\top \\ & \mathbf{c}_2 = -\mathbf{c}_4 = [0, c]^\top & \mathbf{c}_6 = -\mathbf{c}_8 = [-c, c]^\top \\ |\mathbf{c}_0| = 0 & |\mathbf{c}_i| = c \quad i \in \{1, 2, 3, 4\} & |\mathbf{c}_i| = \sqrt{2}c \quad i \in \{5, 6, 7, 8\} \end{array}$$

Les valeurs des vitesses \mathbf{c}_i du schéma D3Q19 sont les suivantes :

$$\begin{array}{lll} \mathbf{c}_0 = [0, 0, 0]^\top & \mathbf{c}_1 = -\mathbf{c}_2 = [c, 0, 0]^\top & \mathbf{c}_7 = -\mathbf{c}_8 = [c, c, 0]^\top \\ & \mathbf{c}_3 = -\mathbf{c}_4 = [0, c, 0]^\top & \mathbf{c}_9 = -\mathbf{c}_{10} = [c, -c, 0]^\top \\ & \mathbf{c}_5 = -\mathbf{c}_6 = [0, 0, c]^\top & \mathbf{c}_{11} = -\mathbf{c}_{12} = [c, 0, c]^\top \\ & & \mathbf{c}_{13} = -\mathbf{c}_{14} = [c, 0, -c]^\top \\ & & \mathbf{c}_{15} = -\mathbf{c}_{16} = [0, c, c]^\top \\ & & \mathbf{c}_{17} = -\mathbf{c}_{18} = [0, c, -c]^\top \\ |\mathbf{c}_0| = 0 & |\mathbf{c}_i| = c \quad i \in \{1, \dots, 6\} & |\mathbf{c}_i| = \sqrt{2}c \quad i \in \{7, \dots, 18\} \end{array}$$

Où c correspond à l'échelle de grandeur des vitesses des particules discrétisées. En unités

LBM, $c = \delta_x / \delta_t = 1$.

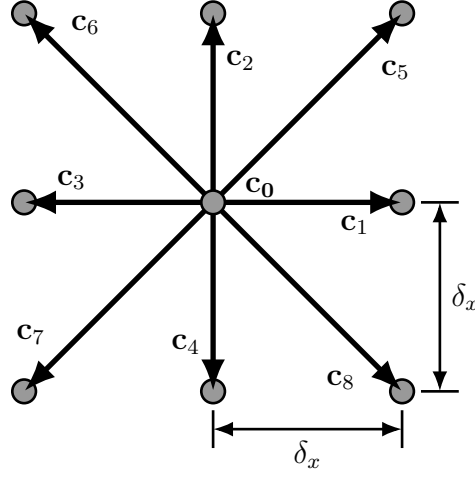


Figure 3.1 Représentation des vecteurs de vitesse \mathbf{c}_i du schéma D2Q9 et de la distance δ_x entre les noeuds

3.1.3 Discrétisation des moments de f

Les intégrales nécessaires au calcul des moments sont évaluées à partir de la quadrature de Gauss-Hermite définie ainsi :

$$\int_{\mathbb{R}^D} \psi(\mathbf{c}) f(\mathbf{x}, \mathbf{c}, t) d^D \mathbf{c} \cong \sum_i \psi(\mathbf{c}_i) w_i f(\mathbf{x}, \mathbf{c}_i, t) \quad (3.3)$$

Où $\psi(\mathbf{c})$ correspond aux différentes fonctions de \mathbf{c} apparaissant dans le calcul des moments. Par exemple, pour le calcul de l'énergie cinétique massique, on retrouve :

$$\rho \epsilon(\mathbf{x}, t) = \int \frac{1}{2} f |\mathbf{c} - \mathbf{u}|^2 d^D \mathbf{c} \cong \sum_i \frac{1}{2} |\mathbf{c}_i - \mathbf{u}|^2 w_i f(\mathbf{x}, \mathbf{c}_i, t) \quad (3.4)$$

Où $\psi(\mathbf{c}) = \frac{1}{2} |\mathbf{c} - \mathbf{u}|^2$. Les poids de discrétisation w_i sont assimilés à la fonction de distribution. Il y a donc une fonction f_i assignée à chacune des vitesses discrétisées \mathbf{c}_i :

$$f_i(\mathbf{x}, t) := w_i f(\mathbf{x}, \mathbf{c}_i, t) \quad (3.5)$$

En substituant l'équation (3.5) dans l'équation (3.3) on obtient alors une quadrature pour l'évaluation des moments de f à partir des f_i .

$$\int_{\mathbb{R}^D} \psi(\mathbf{c}) f(\mathbf{x}, \mathbf{c}, t) d^D \mathbf{c} \cong \sum_i f_i \psi(\mathbf{c}_i) \quad (3.6)$$

On observe que le nombre de vitesses discrétisées \mathbf{c}_i correspond au nombre de points de quadrature. Le schéma DnQm est donc en quelque sorte un schéma de quadrature pour les moments. La masse volumique, la quantité de mouvement et l'énergie cinétique de la fonction de distribution discrétisée f_i sont définies en substituant respectivement les équations (2.19), (2.20) et (2.22) dans la quadrature (3.6).

$$\rho(\mathbf{x}, t) = \sum_i f_i \quad (3.7)$$

$$\rho \mathbf{u}(\mathbf{x}, t) = \sum_i f_i \mathbf{c}_i \quad (3.8)$$

$$\rho \epsilon(\mathbf{x}, t) = \frac{1}{2} \sum_i f_i |\mathbf{c}_i - \mathbf{u}|^2 \quad (3.9)$$

3.1.4 Discrétisation de la fonction de distribution à l'équilibre

La fonction de distribution à l'équilibre f^{eq} est approximée par un polynôme afin que la quadrature impliquée dans l'évaluation de ses moments soit exacte. Ceci permet d'assurer le respect des lois de conservation lors de la discrétisation de l'opérateur BGK. L'ordre minimal nécessaire au polynôme d'approximation de f^{eq} dépend de l'ordre des moments à conserver et nécessite un nombre minimal de points de quadrature. Pour un écoulement isotherme et incompressible, les lois de conservation à respecter sont celles de la masse et de la quantité de mouvement. Puisque la masse volumique et la quantité de mouvement sont des moments d'ordre 0 et 1, l'approximation de f^{eq} doit être au moins d'ordre 2 [18].

La quadrature de Gauss-Hermite est employée afin d'évaluer les moments de la fonction de distribution à l'équilibre :

$$\int_{\mathbb{R}^D} \psi(\mathbf{c}) f^{eq}(\mathbf{x}, \mathbf{c}, t) d^D \mathbf{c} = \sum_i \psi(\mathbf{c}_i) w_i f^{eq}(\mathbf{x}, \mathbf{c}_i, t) \quad (3.10)$$

Les poids de quadrature w_i sont assimilés à f_i^{eq} :

$$f_i^{eq}(\mathbf{x}, t) := w_i f^{eq}(\mathbf{x}, \mathbf{c}_i, t) \quad (3.11)$$

La fonction f^{eq} (2.45) est ensuite développée en une série de polynômes. Pour des raisons d'isotropie, celle-ci est centrée en $\mathbf{u} = 0$ [18] :

$$f^{eq} = \frac{\rho}{(2\pi RT)^{3/2}} \exp \left[\frac{-|\mathbf{c}|^2}{2RT} \right] \left(1 + \frac{\mathbf{c} \cdot \mathbf{u}}{RT} + \frac{(\mathbf{c} \cdot \mathbf{u})^2}{2(RT)^2} - \frac{|\mathbf{u}|^2}{2RT} + O(\mathbf{u}^3) \right) \quad (3.12)$$

Où $|\mathbf{u}|^2 = \mathbf{u} \cdot \mathbf{u}$. La fonction $O(\mathbf{u}^3)$ correspond aux termes d'ordres 3 ou plus. La partie constante de l'équation (3.12) est assimilée au poids d'intégration :

$$W_i := w_i \frac{1}{(2\pi RT)^{3/2}} \exp \left[\frac{-|\mathbf{c}_i|^2}{2RT} \right] \quad (3.13)$$

En introduisant (3.12) et (3.13) dans l'équation (3.11), on obtient la fonction de distribution à l'équilibre discrétisée :

$$f_i^{eq} = W_i \rho \left(1 + \frac{\mathbf{c}_i \cdot \mathbf{u}}{RT} + \frac{(\mathbf{c}_i \cdot \mathbf{u})^2}{2(RT)^2} - \frac{|\mathbf{u}|^2}{2RT} + O(\mathbf{u}^3) \right) \quad (3.14)$$

La série de polynômes est ensuite tronquée en ne conservant que les termes d'ordre 2 ou moins.

$$f_i^{eq} = W_i \rho \left(1 + \frac{\mathbf{c}_i \cdot \mathbf{u}}{RT} + \frac{(\mathbf{c}_i \cdot \mathbf{u})^2}{2(RT)^2} - \frac{|\mathbf{u}|^2}{2RT} \right) \quad (3.15)$$

Puisque l'approximation f_i^{eq} (3.15) est centrée en $\mathbf{u} = \mathbf{0}$, celle-ci est adéquate à faible vitesse $|\mathbf{u}|$ par rapport à l'échelle de vitesse c . On remarque dans la section suivante que l'échelle de vitesse c est directement proportionnelle à la vitesse du son c_s . On conclut alors que l'équation (3.15) est une bonne approximation pour un faible nombre de Mach ($|\mathbf{u}|/c_s$) [19].

3.1.5 Calcul des poids d'intégration

Afin de récupérer les équations de Navier-Stokes dans le cas limite, une attention particulière est nécessaire dans le choix des poids d'intégration w_i . En effet, ceux-ci doivent être choisis en s'assurant que les moments d'ordre 0 à 3 de la fonction f_i^{eq} soient égaux aux moments d'ordre 0 à 3 de la fonction f^{eq} [10]. Ces contraintes s'expriment ainsi :

$$\sum_i f_i^{eq} = \int f^{eq} d^D \mathbf{c} = \rho \quad (3.16)$$

$$\sum_i \mathbf{c}_i f_i^{eq} = \int \mathbf{c} f^{eq} d^D \mathbf{c} = \rho \mathbf{u} \quad (3.17)$$

$$\sum_i \mathbf{c}_i \otimes \mathbf{c}_i f_i^{eq} = \int \mathbf{c} \otimes \mathbf{c} f^{eq} d^D \mathbf{c} \quad (3.18)$$

$$\sum_i \mathbf{c}_i \otimes \mathbf{c}_i \otimes \mathbf{c}_i f_i^{eq} = \int \mathbf{c} \otimes \mathbf{c} \otimes \mathbf{c} f^{eq} d^D \mathbf{c} \quad (3.19)$$

Ces équations peuvent être interprétées comme des contraintes d'isotropie et de respect des lois de conservation locale [10]. La résolution de ce système d'équations, pour un schéma DnQm donné, permet de calculer les poids w_i et de déterminer la relation entre l'échelle des vitesses c et la dispersion de la fonction f^{eq} , c'est-à-dire le produit RT .

La résolution des équations ci-haut pour les schémas D2Q9 et D3Q19 mène aux résultats suivants [20] :

Pour D2Q9 :

$$w_0 = 4/9 \quad (3.20)$$

$$w_{[1,4]} = 1/9 \quad (3.21)$$

$$w_{[5,8]} = 1/36 \quad (3.22)$$

$$c = \sqrt{3RT} \quad (3.23)$$

Pour D3Q19 :

$$w_0 = 1/3 \quad (3.24)$$

$$w_{[1,6]} = 1/18 \quad (3.25)$$

$$w_{[7,18]} = 1/36 \quad (3.26)$$

$$c = \sqrt{3RT} \quad (3.27)$$

3.1.6 L'équation de Boltzmann-BGK discrétisée

La fonction continue f est remplacée par son homologue discrétisé f_i dans l'équation (2.59) à partir des définitions (3.5) et (3.11). :

$$\frac{\partial(f_i/w_i)}{\partial t} + \mathbf{c} \cdot \nabla_{\mathbf{x}}(f_i/w_i) + \mathbf{g} \cdot \nabla_{\mathbf{c}}(f_i/w_i) = \frac{1}{\tau}(f_i^{eq}/w_i - f_i/w_i) \quad (3.28)$$

Pour chaque i , la vitesse \mathbf{c} est une constante qui correspond à \mathbf{c}_i , et le poids d'intégration w_i est une constante simplifiable :

$$\frac{\partial f_i}{\partial t} + \mathbf{c}_i \cdot \nabla_{\mathbf{x}} f_i + \mathbf{g} \cdot \nabla_{\mathbf{c}_i} f_i = \frac{1}{\tau}(f_i^{eq} - f_i) \quad (3.29)$$

Puisque la fonction $f_i(\mathbf{x}, t)$ ne dépend pas de la vitesse \mathbf{c}_i , le troisième terme du membre de gauche est nul. L'effet de la force externe sur l'accélération des particules est plutôt assimilé au terme de collision [21]. L'équation (3.30) correspond donc à l'équation de Boltzmann-BGK discrétisée :

$$\frac{\partial f_i}{\partial t} + \mathbf{c}_i \cdot \nabla_{\mathbf{x}} f_i = \frac{1}{\tau} (f_i^{eq} - f_i) \quad (3.30)$$

3.1.7 Approximation de la dérivée Lagrangienne de f_i

Le membre de gauche de l'équation (3.30) correspond à la dérivée Lagrangienne de f_i . Celle-ci est discrétisée à l'aide d'une différence finie avant d'ordre 1 (*upwind*) :

$$\frac{\partial f_i}{\partial t} + \mathbf{c}_i \cdot \nabla f_i = \frac{df_i(\mathbf{x}, t)}{dt} \cong \frac{f_i(\mathbf{x} + \delta_{\mathbf{x},i}, t + \delta_t) - f_i(\mathbf{x}, t)}{\delta_t} \quad (3.31)$$

Le schéma explicite *upwind* permet une évaluation locale et facilite la parallélisation de l'algorithme. En substituant \mathbf{c}_i à partir de l'équation (3.1) on obtient :

$$\frac{\partial f_i}{\partial t} + \mathbf{c}_i \cdot \nabla f_i \cong \frac{f_i(\mathbf{x} + \mathbf{c}_i \delta_t, t + \delta_t) - f_i(\mathbf{x}, t)}{\delta_t} \quad (3.32)$$

En utilisant les unités LBM, $\delta_t = 1$, on retrouve la forme finale de la dérivée Lagrangienne discrétisée :

$$\frac{\partial f_i}{\partial t} + \mathbf{c}_i \cdot \nabla f_i \cong f_i(\mathbf{x} + \mathbf{c}_i, t + 1) - f_i(\mathbf{x}, t) \quad (3.33)$$

En substituant l'approximation (3.33) à l'équation de Boltzmann-BGK discrétisée (3.30), on trouve l'équation de Boltzmann sur réseau :

$$f_i(\mathbf{x} + \mathbf{c}_i, t + 1) - f_i(\mathbf{x}, t) = \frac{1}{\tau} (f_i^{eq} - f_i) \quad (3.34)$$

Bien que le schéma de différence finie introduit à l'équation (3.31) est d'ordre 1, il est possible de constater que l'équation (3.34) résultante possède une précision d'ordre 2. Ceci est dû au couplage entre le membre de droite et le membre de gauche de l'équation (3.34) [22].

3.1.8 Vitesse du son

La vitesse du son c_s dans un fluide est normalement déterminée ainsi [23] :

$$c_s = \sqrt{\left(\frac{\partial p}{\partial \rho} \right)_s} \quad (3.35)$$

Où l'indice $|_s$ correspond à une contrainte d'entropie constante. Or, il est possible de récupérer l'équation d'état des gaz parfaits pour l'équation de Boltzmann-BGK (2.67). L'équation d'état est également retrouvée en récupérant les équations de Navier-Stokes à partir de l'équation de Boltzmann sur réseau [20]. En substituant l'équation d'état des gaz parfaits dans l'équation (3.35), on retrouve :

$$c_s = \sqrt{\left(\frac{\partial(\rho RT)}{\partial \rho}\right)\bigg|_s} \quad (3.36)$$

Pour un écoulement isotherme, R et T sont constants :

$$c_s = \sqrt{RT \left(\frac{\partial \rho}{\partial \rho}\right)\bigg|_s} = \sqrt{RT} \quad (3.37)$$

Dans le cas d'un écoulement isotherme, le produit RT est remplacé par c_s^2 dans l'équation de f_i^{eq} (3.15). En utilisant les unités LBM, $c = \delta_x/\delta_t = 1$, la vitesse du son c_s pour les schémas D2Q9 et D3Q19 est :

$$c_s = \frac{c}{\sqrt{3}} = \frac{1}{\sqrt{3}} \quad (3.38)$$

Si l'on suppose que le fluide en question se comporte thermodynamiquement comme un gaz parfait, on obtient :

$$c_s = \sqrt{RT} = \sqrt{\gamma RT} \quad (3.39)$$

Où γ est l'indice adiabatique. Ceci implique que $\gamma = 1$, ce qui correspond à un gaz idéal avec une infinité de degrés de liberté.

Dans le contexte de la méthode LB, la vitesse du son est introduite à l'équation d'état (2.67) qui s'exprime ainsi :

$$p = \rho c_s^2 \quad (3.40)$$

3.1.9 Calcul de la viscosité et déduction des équations de Navier-Stokes

Plusieurs méthodes existent afin de récupérer les équations de Navier-Stokes à partir de l'équation de Boltzmann sur réseau [19] [18]. Le passage de LBM à Navier-Stokes permet de retrouver l'équation d'état ainsi que la relation entre le temps de relaxation τ et la viscosité cinématique ν [20] :

$$\nu = c_s^2 (\tau - \delta_t/2) \quad (3.41)$$

La viscosité cinématique pour les schémas D2Q9 et D3Q19 :

$$\nu = \frac{\tau - 0.5}{3} \quad (3.42)$$

Afin de s'assurer d'utiliser une viscosité positive, on déduit une contrainte de stabilité sur le temps de relaxation τ :

$$\tau > 0.5 \quad (3.43)$$

On observe au cinquième cas test qu'une très faible viscosité, pour τ près de 0.5, peut mener à l'instabilité de la méthode LB.

3.1.10 La condition de stabilité de Courant–Friedrichs–Lewy

On pourrait se questionner sur la conséquence de la contrainte de stabilité de Courant sur la méthode LB. Cette contrainte de stabilité s'applique à un schéma de différence finie approximant une équation d'advection (continue). Elle stipule que le domaine de dépendance de la propriété transportée par le schéma de différence finie doit inclure le domaine de dépendance de la propriété transportée par l'équation d'advection continue [24]. Pour le cas de la méthode LB, l'équation continue est l'équation de Boltzmann-BGK discrétisée (3.30) et le schéma de différence finie est l'équation de Boltzmann sur réseau (3.34). Il est possible de vérifier que leurs domaines de dépendances sont identiques.

Selon le schéma de différence finie de (3.34), $f_i(\mathbf{x} + \boldsymbol{\delta}_{\mathbf{x},i}, t + \delta t)$ est calculée à partir de tous les f_i au point (\mathbf{x}, t) (il y a un couplage à travers l'espace des f_i au moment de la collision). Pour la durée d'un pas de temps δt , la contrainte de stabilité de Courant se traduit donc ainsi :

$$\mathbf{x} + \mathbf{c}_i \delta t \leq \mathbf{x} + \boldsymbol{\delta}_{\mathbf{x},i} \quad (3.44)$$

Ceci s'interprète comme quoi, pendant un pas de temps δt , l'information traverse une distance $\boldsymbol{\delta}_{\mathbf{x},i}$ pour le schéma de différence finie, alors qu'elle traverse une distance $\mathbf{c}_i \delta t$ pour l'équation continue. L'information ne doit pas voyager plus rapidement pour l'équation continue que pour le schéma numérique.

La contrainte de stabilité est parfois imposée à partir du nombre de Courant $C \leq 1$ définie ainsi :

$$C = \frac{|\mathbf{c}_i| \delta t}{|\boldsymbol{\delta}_{\mathbf{x},i}|} \leq 1 \quad \forall i \quad (3.45)$$

À partir de l'équation (3.1), on retrouve que pour les schémas D2Q9 et D3Q19 conventionnels le nombre de Courant de la méthode LB est toujours unitaire et respecte donc toujours la contrainte de stabilité. Il est à noter que la contrainte de stabilité de Courant est une condition nécessaire, mais non suffisante à la stabilité d'un schéma d'advection.

Une erreur à éviter lors de l'application de la contrainte de stabilité de Courant à la méthode LB est d'employer la vitesse macroscopique \mathbf{u} comme vitesse advective. Il est important de réaliser que pour la méthode LB, les propriétés transportées sont les f_i et leur vitesse advective sont les \mathbf{c}_i . La vitesse macroscopique \mathbf{u} n'est qu'un des moments des f_i transportés par les f_i . Dans l'équation de la méthode LB (3.34), les moments peuvent intervenir au niveau de la diffusion des f_i , mais n'interviennent pas au niveau de leur advection. En effet, la diffusion (collision) s'effectue dans l'espace des f_i et non pas dans les espaces \mathbf{x} et t .

3.1.11 Système d'équations discrétisées

Le tableau 3.1 décrit le système d'équations permettant de trouver, à partir de l'équation de Boltzmann sur réseau, une approximation de la solution aux équations de Navier-Stokes pour un écoulement isotherme et incompressible. Dans ce système d'équations, W_i , c_s et le nombre de i sont fixés par le choix du schéma DnQm. Le temps de relaxation τ pour sa part permet de contrôler la viscosité cinématique ν . Bien que l'équation de Boltzmann sur réseau soit une équation instationnaire et compressible, l'approximation de f_i^{eq} (3.15) n'est valide que dans un régime quasi incompressible à faible nombre de Mach. Il s'agit donc d'une équation compressible utilisée dans un régime quasi-incompressible.

L'équation Boltzmann-BGK décrit fondamentalement le comportement d'un gaz idéal par le biais de l'utilisation de la distribution de Maxwell-Boltzmann. La méthode LB hérite ainsi de la compressibilité et de l'équation d'état d'un gaz idéal. Les approximations numériques introduites lors de sa discrétisation lui induisent ensuite des propriétés l'écartant du comportement d'un gaz idéal tel que les contraintes d'isothermie et de faible nombre de Mach. Dans le contexte de la simulation d'un fluide incompressible, la faible compressibilité du fluide LB devient donc une source d'erreur numérique.

3.1.12 Conditions aux frontières et conditions initiales

Le rôle des conditions aux frontières et des conditions initiales est d'imposer localement des variables microscopiques, c'est-à-dire des moments de f_i . Il y a plusieurs distributions de particules f_i différentes permettant d'obtenir un même moment. Il s'agit alors de trouver une méthode d'imposition des f_i qui est avantageuse numériquement en terme de précision et

Tableau 3.1 Système d'équations discrétisées

Propriété transportée :	$f_i = f_i(\mathbf{x}, t)$
Équation de transport :	$f_i(\mathbf{x} + \mathbf{c}_i, t + 1) - f_i(\mathbf{x}, t) = C(f_i)$
Condition initiale :	$f_i(\mathbf{x}, t = 0) = f_{i0}$
Loi de comportement :	$C(f_i) = \frac{1}{\tau}(f_i^{eq} - f_i)$ $f_i^{eq} = W_i \rho \left(1 + \frac{\mathbf{c}_i \cdot \mathbf{u}}{c_s^2} + \frac{(\mathbf{c}_i \cdot \mathbf{u})^2}{2c_s^4} - \frac{ \mathbf{u} ^2}{2c_s^2} \right)$
Lois de conservation :	$\rho(\mathbf{x}, t) = \sum_i f_i$ $\mathbf{u}(\mathbf{x}, t) = \frac{1}{\rho} \sum_i f_i \mathbf{c}_i$

de stabilité. Dans le cas des conditions aux frontières de parois solides, il s'agit de modifier l'opérateur de collision afin de représenter la collision des particules avec une paroi solide. Les conditions aux frontières les plus simples à implémenter sont les conditions de *bounce-back* qui permettent de facilement représenter des parois solides *no-slip* ou *free-slip*. Certains modèles de conditions aux frontières sont discutés ci-après.

3.1.13 Algorithme LBM

Lors de l'implémentation, les étapes d'advection et de collision sont séparées :

$$\underbrace{f_i(\mathbf{x} + \mathbf{c}_i, t + 1)}_{\text{advection : } t + 1} = \underbrace{f_i(\mathbf{x}, t) + \frac{1}{\tau}(f_i^{eq} - f_i)}_{\text{collision : } t} \quad (3.46)$$

La matrice f'_i est introduite afin de stocker temporairement les populations de particules entre les étapes d'advection et de collision.

Étape de collision :

$$f'_i(\mathbf{x}, t + 1) = f_i(\mathbf{x}, t) + \frac{1}{\tau}(f_i^{eq} - f_i) \quad (3.47)$$

Étape d'advection :

$$f_i(\mathbf{x} + \mathbf{c}_i, t + 1) = f'_i(\mathbf{x}, t + 1) \quad (3.48)$$

Finalement, voici un aperçu des étapes nécessaires à l'implémentation de l'algorithme LBM, utilisant l'opérateur BGK, pour un écoulement isotherme et incompressible.

Prétraitement :

- Choix de la géométrie : génération du maillage et détermination des conditions aux

frontières

- Choix des paramètres physiques : Nombre de Reynolds
- Choix des paramètres numériques : Finesse du maillage en espace et en temps
- Imposition des conditions initiales

Boucle principale :

- Évaluation des variables macroscopiques (moments) :

$$\rho(\mathbf{x}, t) = \sum_i f_i$$

$$\mathbf{u}(\mathbf{x}, t) = \frac{1}{\rho} \sum_i f_i \mathbf{c}_i$$

- Évaluation de la distribution à l'équilibre :

$$f_i^{eq} = W_i \rho \left(1 + \frac{\mathbf{c}_i \cdot \mathbf{u}}{c_s^2} + \frac{(\mathbf{c}_i \cdot \mathbf{u})^2}{2c_s^4} - \frac{|\mathbf{u}|^2}{2c_s^2} \right)$$

- Collision :

$$f'_i(\mathbf{x}, t) = f_i(\mathbf{x}, t) + \frac{1}{\tau} (f_i^{eq} - f_i)$$

Imposition des conditions aux frontières

- Advection :

$$f_i(\mathbf{x} + \mathbf{c}_i, t + 1) = f'_i(\mathbf{x}, t)$$

Post-traitement :

- Récupération de la solution : champ tensoriel $f_i(\mathbf{x}, t)$
- Les variables macroscopiques sont évaluées à partir des moments de f_i et de l'équation d'état $p = \rho c_s^2$.

La boucle principale peut être facilement parallélisée lors de son implémentation. En effet, l'évaluation des moments, l'évaluation de la distribution à l'équilibre et l'étape de collision peuvent être évaluées de manière indépendante pour chaque noeud.

3.2 Implémentation

Dans la présente section, la librairie *open-source* Palabos est brièvement présentée. Par la suite, le choix des paramètres numériques employés est justifié.

3.2.1 La librairie *open-source* Palabos

Palabos consiste en une implémentation de l'algorithme LBM sous forme de librairie C++ *open-source*. L'utilisation de cette librairie permet de profiter d'une structure de programmation bien développée, optimisée et parallélisée pour le calcul sur plusieurs processeurs. Le caractère *open-source* permet l'accès à l'entièreté de la structure de programmation, ce qui rend possible l'intégration de nouveaux modèles. Plusieurs modèles basés sur la méthode LB sont implémentés dans la librairie Palabos, tels que des modèles pour écoulements : thermiques, turbulents, multiphasés, avec objets mobiles immergés, avec interactions fluide-

particules, etc. [25]

Tous les opérateurs de collisions, les modèles de conditions aux frontières, et les autres fonctions rattachées au fonctionnement de l'algorithme LBM utilisés sont déjà implémentés dans la librairie Palabos. Aucun nouveau modèle n'a été implémenté. Le travail de programmation effectuée se résume plutôt à la génération de la géométrie, au calcul des paramètres intrants et extrants, et à l'utilisation des modèles déjà implémentés. Ci-après sont présentés les opérateurs de collision et les modèles de conditions aux frontières implémentés dans Palabos et utilisés dans le cadre des cas tests étudiés.

3.2.2 Les opérateurs de collision

L'opérateur BGK a déjà été abordé dans les chapitres précédents. Il est le premier modèle employé à l'origine de la méthode LBM et demeure le plus simple. Différents opérateurs collision permettent entre autres d'augmenter la stabilité de la méthode au coût de l'augmentation du temps de calcul.

L'opérateur MRT

L'opérateur à temps de relaxation multiples (*Multiple-Relaxation-Time*) est détaillé dans la référence [9]. C'est une méthode qui consiste à remplacer le temps de relaxation τ de l'opérateur BGK par une matrice appelée matrice de collision. La matrice de collision est de dimension $Q \times Q$ où Q correspond au nombre de différentes vitesses i de la fonction de distribution f_i . Celle-ci est diagonalisée afin de l'exprimer à partir d'un nombre Q de différents temps de relaxation. Elle est construite de manière à ce que chacun de ses vecteurs propres soit associé au retour à l'équilibre d'un moment. Chaque valeur propre correspond pour sa part à l'inverse du temps de relaxation vers l'équilibre d'un moment. Ceci permet donc, lors de l'étape de collision, de contrôler de manière indépendante la relaxation vers l'équilibre de Q moments de f_i à partir de Q temps de relaxation. Pour des écoulements non thermiques, les seuls moments qui ont une pertinence physique et qui sont sujets aux lois de conservation sont ceux permettant d'exprimer la densité, la vitesse et le tenseur de contraintes. Les autres moments n'ont aucune pertinence physique et correspondent donc à des paramètres numériques. La matrice de collision du modèle MRT est construite de manière à retrouver les équations de Navier-Stokes dans le cas limite, et favoriser la précision et la stabilité de la méthode. Le modèle BGK est entre autres un cas particulier du modèle MRT pour lequel tous les temps de relaxation sont égaux.

L'opérateur BGK Régularisé

Le modèle BGK Régularisé est détaillé dans la référence [26]. C'est une variante du modèle BGK pour laquelle on ajoute une étape de régularisation de f_i entre l'étape d'advection et l'étape de collision. L'étape de régularisation consiste à réexprimer la fonction de distribution en éliminant complètement les moments qui n'ont pas de pertinence physique tout en conservant les moments qui ont une pertinence physique (densité, vitesse, tenseur de contraintes). Il est possible d'assimiler l'étape de régularisation à une matrice de collision afin d'exprimer la méthode BGK-Régularisé de manière similaire à la méthode MRT. Les deux méthodes diffèrent alors dans leur manière de déterminer les temps de relaxation associés aux différents moments. Pour la méthode MRT, les moments non physiques sont relaxés vers l'équilibre afin de maximiser la stabilité et la précision, alors que pour la méthode BGK-Régularisé, ces moments non physiques sont tout simplement éliminés.

L'opérateur BGK avec force externe

Le modèle BGK avec force externe employée est celui de Guo détaillé dans la référence [21]. Un terme de force externe est ajouté au membre de droite de l'équation de Boltzmann sur réseau avec modèle BGK. Ce terme est ensuite développé en une série de puissance d'ordre 2, puis la procédure de Chapman-Enskog est appliquée; les constantes apparaissant dans le développement du terme de force externe sont déterminées en s'assurant de récupérer les équations de Navier-Stokes dans le cas limite. Des corrections sont également apportées à l'expression de distribution à l'équilibre f_i^{eq} et de la vitesse macroscopique \mathbf{u} .

L'opérateur MRT avec force externe

Le modèle MRT avec force externe employée est détaillé dans la référence [27]. Il s'agit d'inclure le terme de force externe de Guo cité ci-haut à l'opérateur de collision MRT. On applique la procédure de Chapman-Enskog à l'équation de Boltzmann sur réseau avec modèle MRT auquel est ajouté le terme de force externe d'ordre 2.

3.2.3 Les modèles de conditions aux frontières

La plupart des conditions aux frontières employées dans les cas tests traités ci-après concernent des frontières rectilignes. Seule une méthode de type *bounce-back* est employée pour représenter certaines frontières courbes. Ces frontières courbes sont représentées par des surfaces en forme d'escaliers, c'est-à-dire par des petits éléments de surfaces perpendiculaires soit à l'axe des x , y ou z . La frontière *bounce-back* est également la seule condition à la frontière

employée qui conserve la masse de manière exacte. Pour une cellule à la frontière du domaine, certaines populations de particules sont inconnues suite à l'étape d'advection. On entend par population un des différents f_i d'une cellule donnée. Les populations inconnues sont celles qui devraient provenir de cellules à l'extérieur du domaine de simulation lors de l'étape d'advection. Certaines conditions aux frontières remplacent uniquement les populations inconnues, alors que certaines autres les remplacent toutes. Les méthodes remplaçant uniquement les populations inconnues sont généralement plus précises alors que les méthodes qui remplacent toutes les populations sont généralement plus stables [28]. Bien que certaines méthodes peuvent être plus précises, celles présentées ci-dessous peuvent atteindre dans certains cas une précision d'ordre 2 sur le champ de vitesse [28] [29]. Les méthodes plus stables sont plus avantageuses pour des écoulements à nombre de Reynolds élevé.

Frontière *bounce-back*

La méthode *bounce-back* employée est décrite dans la référence [11]. Il est à noter que l'auteur fait la distinction entre deux types de frontières *bounce-back*, soit la méthode *on-grid bounce-back* et la méthode *mid-grid bounce-back*. Dans le présent document, seule la méthode *mid-grid bounce-back* est employée et référée comme étant la méthode *bounce-back*. Cette condition de frontière sans glissement consiste à réfléchir les flux de particules incidents sur la paroi vers la direction d'où ils proviennent. Il s'agit d'un schéma centré en espace et en temps qui permet d'obtenir dans certains cas une précision d'ordre 2 sur la vitesse [29].

Frontière de Skordos

La méthode de Skordos est décrite dans la référence [30]. La condition à la frontière de Skordos remplace toutes les populations de particules. L'opérateur de collision est modifié à la frontière afin que la fonction de distribution à l'équilibre soit ajustée en fonction du gradient de vitesse macroscopique local. Le gradient de vitesse est évalué à partir d'un schéma de différences finies.

Frontière d'Inamuro

La méthode d'Inamuro est décrite dans la référence [31]. Les populations de particules inconnues sont déterminées à partir d'une fonction de distribution à l'équilibre f_{eq} légèrement décalée. Ce décalage est effectué au niveau de la vitesse macroscopique de sorte que la vitesse macroscopique des particules à la paroi soit la même que la vitesse de la paroi.

Frontière de Zou-He

La méthode Zou-He est décrite dans la référence [29]. Seules les populations inconnues sont calculées à partir d'un schéma inspiré de la condition *bounce-back*. On suppose que la partie non à l'équilibre des populations de particules est réfléchie.

Frontière régularisée

La méthode de frontière régularisée est décrite dans la référence [32]. La méthode régularisée remplace toutes les populations de particules, incluant celles qui sont connues. Le tenseur de contraintes est d'abord évalué à partir des populations de particules connues. Une étape de régularisation est ensuite effectuée, de manière similaire à l'opérateur de collision régularisé mentionné ci-haut.

Dans le chapitre suivant, plusieurs de ces différents modèles sont vérifiés sur des cas tests typiques. Bien que plusieurs modèles DnQm soient implémentés, seul les modèles D2Q9 et D3Q19 sont employés pour les écoulements en 2D et 3D respectivement.

3.2.4 Paramètres de simulation d'écoulement incompressible

Les paramètres N_L et N_t permettent de contrôler la résolution du maillage en espace et en temps respectivement. Ils correspondent au nombre de sous-intervalles (cellules), entre chaque noeud, permettant de représenter une longueur de référence L_0 et une durée de référence t_0 respectivement. On calcule de la manière suivante la distance entre deux noeuds en espace δ_x et en temps δ_t :

En unités LBM :

$$\delta_x = L_0/N_L \quad (3.49)$$

$$\delta_t = t_0/N_t \quad (3.50)$$

En unités adimensionnelles :

$$\tilde{\delta}_x = \delta_x/L_0 = 1/N_L \quad (3.51)$$

$$\tilde{\delta}_t = \delta_t/t_0 = 1/N_t \quad (3.52)$$

La distance entre deux noeuds en espace et en temps est implémenté comme étant égale à 1 dans l'algorithme LBM :

$$\delta_x = \delta_t = 1 \quad (3.53)$$

À partir des équations (3.51) et (3.52) et (3.53), on déduit :

$$N_L = L_0 \quad (3.54)$$

$$N_t = t_0 \quad (3.55)$$

La vitesse de référence U_0 en unités LBM est donc :

$$U_0 = \frac{L_0}{t_0} = \frac{N_L}{N_t} = \frac{\tilde{\delta}_t}{\tilde{\delta}_x} \quad (3.56)$$

Le nombre de Reynolds est introduit et les équations (3.51) et (3.52) y sont substituées, pour une viscosité constante $\nu_0 = \nu$:

$$\text{Re} = \frac{U_0 L_0}{\nu_0} = \frac{N_L^2}{N_t \nu} = \frac{\tilde{\delta}_t}{\tilde{\delta}_x^2 \nu} \quad (3.57)$$

Pour une simulation donnée, la résolution spatiale N_L est déterminée lors de la génération du maillage spatial. La résolution temporelle N_t pour sa part peut être déterminée soit en fixant le temps caractéristique t_0 , la vitesse caractéristique U_0 , la viscosité ν ou le temps de relaxation τ . La différence entre ces choix est apparente lors du raffinement de maillage spatial (augmentation de N_L). Au tableau 3.2, on observe la dépendance du nombre de Mach par rapport à N_L pour chacun des paramètres mentionnés ci-haut gardés constants. On observe que pour une résolution spatiale qui augmente, fixer t_0 provoque une augmentation du nombre de Mach, fixer U_0 maintient le nombre de Mach constant, alors que fixer ν ou τ provoque une diminution du nombre de Mach. L'avantage de fixer ν ou τ est que l'erreur de compressibilité est réduite du même ordre que l'erreur due à la discrétisation spatiale lorsque le maillage est raffiné. [33]

Enfin, pour un Re et un N_L donné, les actions suivantes ont toutes l'effet d'augmenter la résolution temporelle N_t : diminuer le pas de temps adimensionnel $\tilde{\delta}_t$, diminuer la vitesse caractéristique U_0 , diminuer le temps de relaxation τ , diminuer la viscosité ν . Pour chacun des cas tests abordés, un tableau de paramètres décrit comment la résolution temporelle N_t est déterminée.

La masse volumique caractéristique ρ_0 employée est unitaire pour tous les cas tests abordés, soit $\rho_0 = 1$.

Tableau 3.2 Effet du paramètre contrôlant la résolution temporelle sur la dépendance à la résolution spatiale N_L du nombre de Mach

Paramètre constant	N_t	Ma
t_0	$N_t = t_0$	$\text{Ma} = \frac{N_L}{t_0 c_s^2}$
U_0	$N_t = \frac{N_L}{U_0}$	$\text{Ma} = \frac{U_0}{c_s^2}$
ν	$N_t = \frac{N_L^2}{\text{Re } \nu}$	$\text{Ma} = \frac{\nu \text{Re}}{c_s^2 N_L}$
τ	$N_t = \frac{N_L^2}{\text{Re } c_s^2 (\tau - 1/2)}$	$\text{Ma} = \frac{(\tau - 1/2) \text{Re}}{N_L}$

3.2.5 La loi de Darcy

Parmi les cas tests étudiés dans les chapitres suivants, deux concernent l'utilisation de la méthode LB afin de calculer la perméabilité de milieux poreux. La loi de Darcy permet de caractériser le comportement macroscopique d'un écoulement à travers un milieu poreux. Un milieu poreux à géométrie complexe est alors modélisé à partir d'un simple paramètre de perméabilité. La loi de Darcy est définie ainsi [34] :

$$\mathbf{q} = -\frac{\kappa}{\mu} (\nabla p + \rho \mathbf{g}) \quad (3.58)$$

Où le flux de Darcy \mathbf{q} correspond au flux volumique moyen d'un fluide incompressible à travers un élément de milieu poreux δV :

$$\mathbf{q} = \frac{\int_{\delta V} \mathbf{u} \, d\mathbf{x}}{\int_{\delta V} d\mathbf{x}} \quad (3.59)$$

Le paramètre κ correspond à la perméabilité du milieu poreux. Pour le présent cas d'un milieu isotrope, ce paramètre est un scalaire. Le flux de Darcy est lié à la vitesse moyenne volumique du fluide $\bar{\mathbf{u}}$ par l'intermédiaire de la porosité ϕ :

$$\mathbf{q} = \phi \bar{\mathbf{u}} \quad (3.60)$$

Où la porosité ϕ correspond à la proportion de volume occupée par tous les pores V_{pores} :

$$\phi = \frac{V_{pores}}{V}$$

Le flux de Darcy \mathbf{q} est évalué sur un élément de volume δV du milieu poreux de taille suffisamment grande par rapport à la taille des pores du milieu. Bien que l'équation de Darcy soit empirique, il est possible d'observer son lien avec les équations de transport en introduisant certaines hypothèses simplificatrices. Étant donné qu'un milieu poreux est constitué d'un réseau complexe de petits canaux, les forces de frottement avec les surfaces sont importantes à travers l'écoulement. Il s'agit donc d'un écoulement rampant où les forces visqueuses sont dominantes. On rappelle tout d'abord l'équation de transport de quantité de mouvement (2.37) :

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) = -\nabla p + \nabla \cdot \boldsymbol{\tau}_{ij} + \mathbf{f} \quad (3.61)$$

Où la force externe \mathbf{f} est l'accélération gravitationnelle $\rho \mathbf{g}$:

$$\mathbf{f} = \rho \mathbf{g} \quad (3.62)$$

On suppose un écoulement de fluide incompressible, stationnaire et à faible nombre de Reynolds. L'hypothèse de faible nombre de Reynolds permet de négliger le membre de gauche de l'équation (3.61) qui représente la contribution des forces d'inertie :

$$-\nabla p + \nabla \cdot \boldsymbol{\tau}_{ij} + \rho \mathbf{g} = 0 \quad (3.63)$$

On observe ensuite une hypothèse intrinsèque à l'équation empirique de Darcy, soit que les forces visqueuses sont proportionnelles au flux macroscopique \mathbf{q} et la viscosité cinétique μ du fluide :

$$\nabla \cdot \boldsymbol{\tau}_{ij} \equiv -\frac{\mu}{\kappa} \mathbf{q} \quad (3.64)$$

La perméabilité κ pour sa part est un paramètre caractérisant la géométrie du milieu. Une perméabilité plus élevée correspond à une géométrie offrant une plus faible résistance à l'écoulement. La loi de Darcy est similaire à la loi de Stokes dans le sens où les deux lois décrivent un lien de proportionnalité entre les forces visqueuses d'un fluide newtonien rampant et une vitesse caractéristique de l'écoulement.

On retient que la loi de Darcy devrait bien représenter les écoulements rampants de fluides newtoniens, stationnaires et incompressibles. Dans le cas où ces hypothèses ne sont pas respectées, il serait tout de même possible d'employer la loi de Darcy pour modéliser la perméabilité. Cependant, la perméabilité κ ne serait plus nécessairement constante.

3.2.6 Vérification de la méthode LB sur divers cas test

Pour un écoulement incompressible et stationnaire, la solution obtenue par la méthode LB dépend des paramètres physiques suivants : la géométrie, les conditions aux frontières, le nombre de Reynolds ; et des paramètres numériques suivants : la condition initiale, l'opérateur de collision, les modèles de conditions aux frontières et de condition initiale, la résolution en espace N_L et le temps de relaxation τ . Ici on distingue condition initiale et modèle de condition initiale ; et on distingue conditions aux frontières et modèles de conditions aux frontières. Par conditions aux frontières on entend le sens physique tel qu'imposer la pression, la vitesse ou une périodicité. Par modèles de conditions aux frontières, on entend les schémas numériques employés pour imposer la condition physique, tel que la frontière Zou-He ou la frontière *bounce-back*. La dépendance de la solution obtenue par rapport à la plupart des paramètres décrits ci-haut est étudiée dans le cadre des cas tests présentés aux chapitres suivants. Quelques-uns des paramètres employés pour une ou plusieurs simulations sont regroupés dans des tableaux récapitulatifs.

CHAPITRE 4 RÉSULTATS DE SIMULATIONS NUMÉRIQUES

4.1 Écoulement de Poiseuille 2D entre deux plaques

Le premier cas étudié est l'écoulement de Poiseuille 2D entre deux plaques planes. La figure 4.1 décrit le domaine de simulation. Sur celle-ci l_y est la distance entre les deux plaques et l_x est la longueur du domaine. Puisque la longueur du domaine est infinie dans le cas analytique, l_x est considéré comme un paramètre numérique. La solution analytique obtenue à partir des équations de Navier-Stokes incompressibles permet de déterminer le champ de vitesses analytique $\mathbf{u}_a = [u_a, v_a]^\top$ et le champ de pression analytique p_a . Les composantes du champ de vitesse analytique sont exprimées aux équations (4.1) et (4.2) [23].

$$u_a(y) = U_{max} \left(1 - 4 \left(\frac{y}{l_y} \right)^2 \right) \quad (4.1)$$

$$v_a = 0 \quad (4.2)$$

Où U_{max} est la vitesse maximale du profil de vitesse parabolique. La méthode LB est quasi incompressible et la pression est imposée par le biais de la masse volumique à partir de la relation (3.40). La masse volumique caractéristique $\rho_0 = 1$ est fixée à mi-chemin ($x = l_x/2$) de l'écoulement.

La pression analytique pour un écoulement de Poiseuille 2D incompressible est exprimée ainsi :

$$p_a(x) = p_a(l_x/2) + \frac{8 \nu \rho U_{max}}{l_y^2} \left(\frac{l_x}{2} - x \right) \quad (4.3)$$

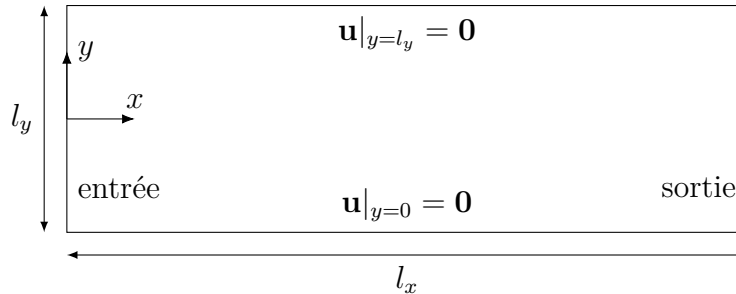


Figure 4.1 Schéma du domaine de calcul pour l'écoulement de Poiseuille en 2D entre deux plaques

Où la pression à mi-chemin de l'écoulement est fixée à $\rho(x = l_x/2) = \rho_0 = 1$:

$$p_a(x = l_x/2) = \rho_0 c_s^2 = c_s^2 \quad (4.4)$$

Les conditions initiales sont fixées à vitesse nulle et masse volumique unitaire :

$$\mathbf{u}|_{t=0} = \mathbf{0} \quad (4.5)$$

$$\rho|_{t=0} = \rho_0 = 1 \quad (4.6)$$

La condition de non-glissement est appliquée aux deux plaques planes :

$$\mathbf{u}|_{y=0} = \mathbf{u}|_{y=l_y} = \mathbf{0} \quad (4.7)$$

Deux types de conditions à l'entrée et à la sortie sont comparées. Dans le premier cas, la vitesse est imposée à l'entrée et à la sortie. Dans le second cas, la pression est imposée à l'entrée et à la sortie. Pour le cas de vitesse imposée, les équations (4.1) et (4.2) permettent de déterminer le profil à imposer à l'entrée et à la sortie :

$$\mathbf{u}|_{x=0} = \mathbf{u}|_{x=l_x} = U_{max} \left(1 - 4 \left(\frac{y}{l_y} \right)^2 \right) \mathbf{i} + 0\mathbf{j} \quad (4.8)$$

Où \mathbf{i} et \mathbf{j} correspondent aux vecteurs unitaires dans le sens des axes x et y respectivement. Pour le cas de pression imposée, l'équation (4.3) permet de déterminer le profil à imposer à l'entrée et à la sortie :

$$\rho|_{x=0} = \frac{p|_{x=0}}{c_s^2} = 1 + \frac{8 \nu \rho U_{max} l_x}{l_y^2 c_s^2} \frac{1}{2} \quad (4.9)$$

$$\rho|_{x=l_x} = \frac{p|_{x=l_x}}{c_s^2} = 1 - \frac{8 \nu \rho U_{max} l_x}{l_y^2 c_s^2} \frac{1}{2} \quad (4.10)$$

Pour ce cas test, le modèle de condition à la frontière Régularisé est employée pour toutes les frontières. Ce modèle, qui est avantageux pour sa stabilité, est utilisé en prévision du prochain cas test de la cavité carré, dont les simulations sont effectuées à haut nombre de Reynolds. Le nombre de Reynolds est évalué en considérant la vitesse maximale U_{max} et la distance entre les plaques l_y comme vitesse caractéristique $U_0 = N_L/N_t$ et longueur caractéristique L_0 :

$$\text{Re} = \frac{U_0 L_0}{\nu_0} = \frac{U_{max} l_y}{\nu} \quad (4.11)$$

La qualité des résultats de simulations est évaluée en mesurant la distance entre la solution analytique et la solution numérique obtenue. Cette distance est calculée à partir de l'erreur quadratique moyenne $E_{\mathbf{u}}$ de l'approximation numérique par rapport à la solution analytique. L'erreur $E_{\mathbf{u}}$ est calculée sur le champ de vitesses et elle est définie par l'équation (4.12).

$$E_{\mathbf{u}} = \frac{1}{U_0} \sqrt{\frac{1}{k} \sum_i \left((u_{n,i} - u_{a,i})^2 + (v_{n,i} - v_{a,i})^2 \right)} \quad (4.12)$$

Où $\mathbf{u}_{n,i} = [u_{n,i}, v_{n,i}]^T$ et $\mathbf{u}_{a,i} = [u_{a,i}, v_{a,i}]^T$ correspondent respectivement aux valeurs numériques et analytiques de la vitesse au noeud i , et k correspond au nombre total de noeuds. L'erreur sur le champ de pression est également employée, celle-ci est définie à partir de l'erreur relative moyenne :

$$E_p = \frac{1}{k} \sum_i \frac{|p_{n,i} - p_{a,i}|}{p_0} \quad (4.13)$$

Où $p_{n,i}$ et $p_{a,i}$ correspondent respectivement aux valeurs numériques et analytiques de la pression au noeud i . La pression $p_0 = \rho_0 c_s^2 = 1/3$ est la pression caractéristique. La pression analytique est déterminée à partir de l'équation (4.3) et la pression numérique est évaluée à partir du champ de densité à partir de l'équation d'état $p = \rho c_s^2$.

Pour chaque simulation, la condition de régime permanent est vérifiée en évaluant la différence de l'erreur $E_{\mathbf{u}}$ calculée entre deux pas de temps successifs. La simulation est arrêtée lorsque cette différence est inférieure à 10^{-16} .

4.1.1 Effet de la longueur du domaine sur l'erreur numérique

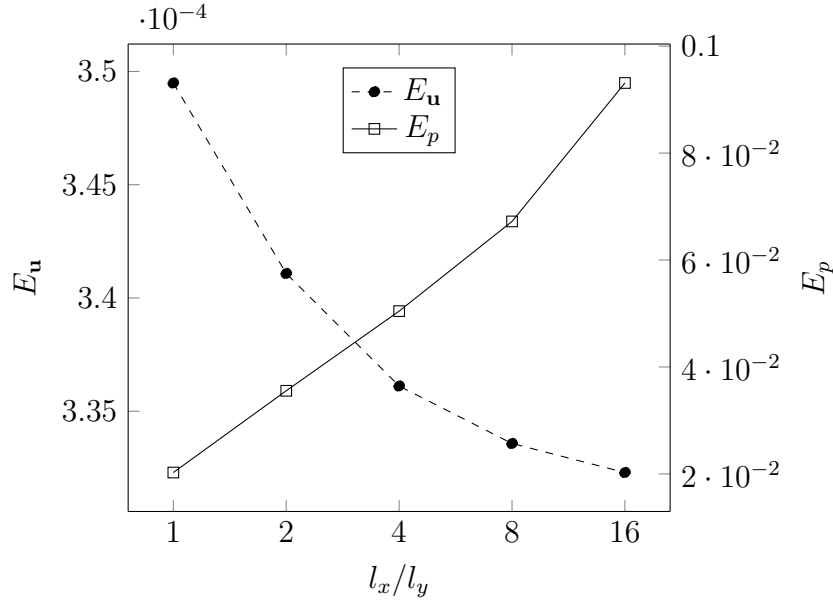
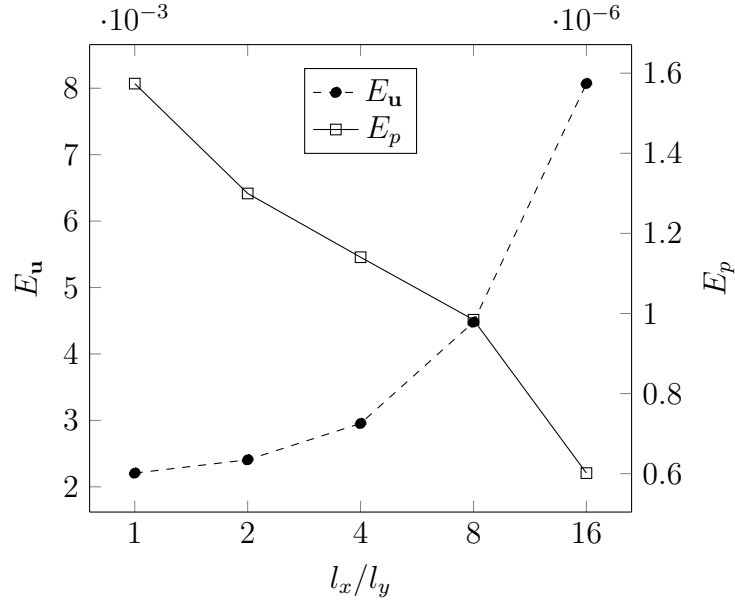
Les paramètres de simulation utilisés sont résumés au tableau 4.2a où le pas de temps est évalué pour une viscosité $\nu = 0.1$. Dans les graphiques de la figure 4.2, on observe l'effet de la longueur du domaine l_x sur les erreurs $E_{\mathbf{u}}$ et E_p pour deux types de conditions à l'entrée et à la sortie. Les deux types de conditions sont soit d'imposer la pression à l'entrée et à la sortie ou d'imposer la vitesse à l'entrée et à la sortie. Le but est d'observer l'effet sur l'erreur causée par la présence des frontières et l'approximation d'un domaine infini par un domaine fini. Plus le domaine est long, plus petit est le nombre de cellules attribuées aux conditions aux frontières pour une même résolution spatiale.

On remarque que pour la pression imposée, le champ de vitesse s'éloigne de la solution analytique lorsque le domaine de simulation est plus long. De manière réciproque, pour la vitesse imposée, le champ de pression s'éloigne de la solution analytique lorsque le domaine de

Paramètre

Nombre de Reynolds (Re)	10
Modèle(s) de collision	BGK
Modèle(s) de condition aux frontières	Régularisé
Résolution spatiale (N_L)	$N_L = l_y = 32$
Résolution temporelle (N_t)	$N_t _{\nu=0,1} = \frac{N_L^2}{\text{Re} \nu} = N_L^2$

(a) Paramètres de simulation pour la section 4.1.1.

(b) Erreur E_u en fonction de l_x/l_y avec vitesse imposée.(c) Erreur E_p en fonction de l_x/l_y avec pression imposée.Figure 4.2 Erreur en fonction de l_x/l_y pour deux types de conditions aux frontières

simulation est plus long. On rappelle que la solution analytique correspond à un écoulement parfaitement incompressible. Ceci met en évidence la faible compressibilité de la méthode LBM. Une augmentation de la longueur du domaine l_x provoque un plus grand rapport de pression entre l'entrée et la sortie, ce qui accroît l'écart entre une solution compressible et incompressible. Bien que le nombre de Mach demeure constant lorsque la de la longueur du domaine varie, les effets de la compressibilité deviennent quand même plus importants.

Pour le cas de la vitesse imposée à l'entrée et à la sortie, étant donné que la méthode est faiblement compressible, le champ de pression n'atteint pas de régime permanent. Puisque la pression à l'entrée est plus élevée qu'à la sortie, et que le même profil de vitesse supposant un écoulement incompressible est imposé, il y a une constante accumulation de masse dans le domaine. La pression est calculée à partir de la densité, ce qui provoque une continuelle augmentation de la pression. Ceci n'est pas le cas pour ce qui est de la pression imposée où les champs de vitesse et de pression se stabilisent rapidement.

4.1.2 Effet du modèle de conditions aux frontières sur l'erreur numérique

Les paramètres de simulation utilisés sont résumés au tableau 4.3a où la résolution temporelle N_t est évaluée pour une viscosité $\nu = 0.1$. Pour cette simulation, la vitesse est imposée à l'entrée et à la sortie. Le graphique de la figure 4.3b montre l'effet de différentes conditions aux frontières sur l'erreur. On observe que les deux modèles les plus précis sont ceux qui conservent les populations de particules, soit les modèles Inamuro et Zou-He dont les courbes sont confondues. L'effet de conserver les populations connues aux frontières permet d'obtenir une plus grande précision en maximisant l'information provenant de l'écoulement, plutôt que celle provenant de la frontière [28]. Le modèle BGK ainsi que les conditions aux frontières utilisées sont des méthodes pouvant atteindre une convergence d'ordre de 2 [28], ce qui coïncide avec les résultats obtenus.

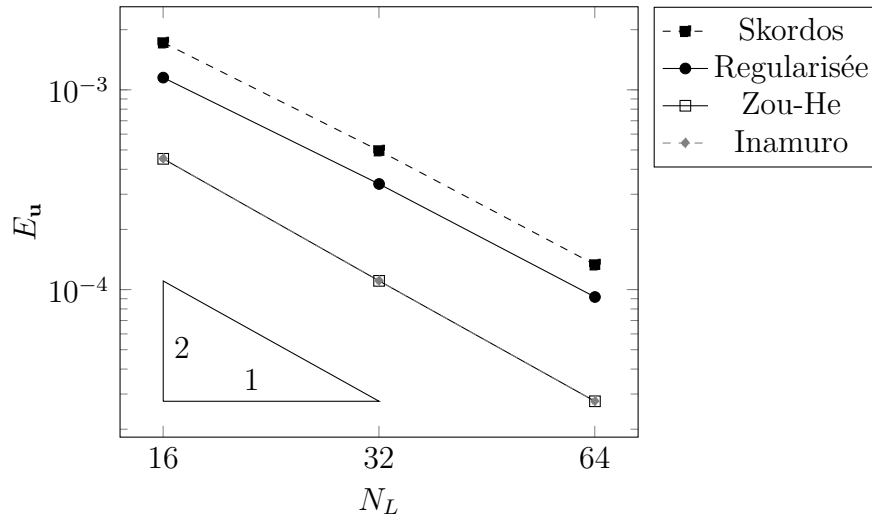
4.1.3 Effet du nombre de Reynolds sur l'erreur numérique

Les paramètres de simulation utilisés sont résumés au tableau 4.4a où la résolution temporelle N_t est évaluée pour une vitesse $U_0 = 2^{-6}$. Pour cette simulation, la vitesse est imposée à l'entrée et à la sortie. Dans la figure (4.4b) on observe l'effet du nombre de Reynolds sur l'erreur. On remarque qu'il y a un pic là où l'erreur est la plus faible, soit $Re = 6$. Ceci correspond à un temps de relaxation unitaire $\tau = 1$. L'erreur augmente pour des Re plus faibles ou plus élevés.

Paramètre

Nombre de Reynolds (Re)	10
Modèle(s) de collision	BGK
Modèle(s) de condition aux frontières	Régularisé, Skordos, Zou-He, Inamuro
Résolution spatiale (N_L)	$N_L = l_y = \{16, 32, 64\}$
Résolution temporelle (N_t)	$N_t _{\nu=0,1} = \frac{N_L^2}{\text{Re} \nu} = N_L^2$

(a) Paramètres de simulation pour la section 4.1.2.

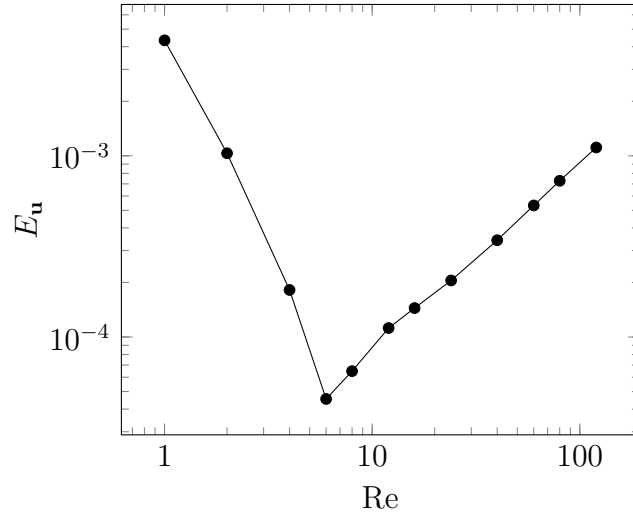


(b) Erreur en fonction de la résolution pour $l_x/l_y = 3$ et $\text{Re} = 10$.

Figure 4.3 Effet du modèle de conditions aux frontières sur l'erreur numérique E_u

Paramètre	
Nombre de Reynolds (Re)	$[1, 120]$
Modèle(s) de collision	BGK
Modèle(s) de condition aux frontières	Régularisé
Résolution spatiale (N_L)	$N_L = l_y = 64$
Résolution temporelle (N_t)	$N_t _{U=2^{-6}} = \frac{N_L}{U} = 4096$

(a) Paramètres de simulation pour la section 4.1.3.

(b) Erreur en fonction du nombre de Reynolds pour $l_x/l_y = 3$.Figure 4.4 Effet du nombre de Reynolds sur l'erreur numérique E_u

4.1.4 Effet de la résolution sur le temps de calcul

Les paramètres de simulation utilisés sont résumés au tableau 4.5a. Pour cette simulation, la vitesse est imposée à l'entrée et à la sortie. Le but est d'évaluer s'il est préférable de raffiner la résolution en espace ou en temps afin d'obtenir le compromis le plus avantageux entre précision et temps de calcul. On définit le nombre de sites mis à jour pendant la simulation de la manière suivante :

$$\text{Nombre de sites mis à jour} = (\text{Nombre de noeuds}) \cdot (\text{Nombre de pas de temps}) \quad (4.14)$$

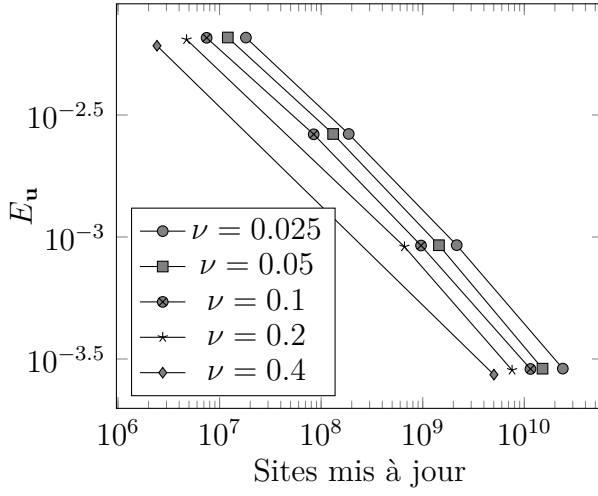
Les opérations élémentaires de la boucle principale de l'algorithme LBM sont répétées pour chaque site mis à jour. Le nombre de sites mis à jour est donc proportionnel au temps de calcul. Lorsque $Re = 10$, on remarque à la figure 4.5b que pour un même nombre de sites mis à jour, il serait préférable de raffiner en espace plutôt qu'en temps. Lorsque $Re = 1$, on remarque à la figure 4.5c que pour un même nombre de sites mis à jour, il serait préférable de raffiner en temps plutôt qu'en espace. Pour $Re = 0.1$, on remarque à la figure 4.5d que pour un même nombre de sites mis à jour, la viscosité $\nu = 10$ serait la plus avantageuse. Dans ce cas, il n'est pas nécessairement préférable de prioriser un raffinement en temps ou en espace. On conclut alors que pour un même nombre de sites mis à jour, le rapport entre la résolution en espace et en temps le plus avantageux devrait dépendre du nombre de Reynolds. Finalement, il est important de préciser que la parallélisation de l'algorithme se fait en espace et non en temps, il serait alors préférable de privilégier un raffinement en espace plutôt qu'en temps pour un même nombre de sites mis à jour.

Un des phénomènes observés lors de l'étude de ce cas test est la variation constante de la masse totale de l'écoulement. Ce phénomène est donc approfondi dans le cadre du prochain cas test. Un système fermé, soit la cavité carrée avec paroi entraînée, est choisi afin de s'assurer que la variation de masse observée n'est pas uniquement due à la compressibilité de la méthode.

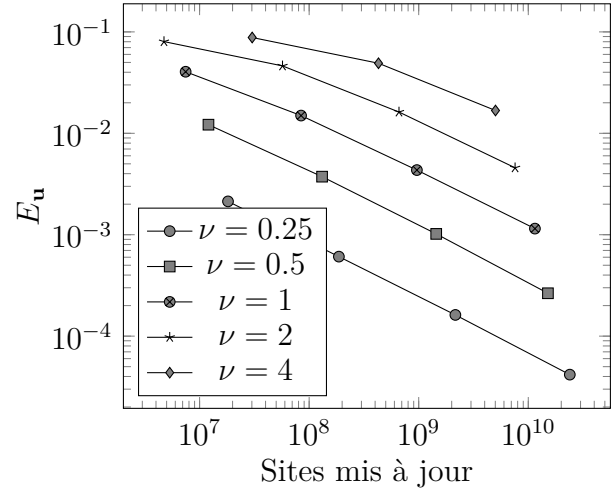
Paramètre

Nombre de Reynolds (Re)	0.1, 1, 10
Modèle(s) de collision	BGK
Modèle(s) de condition aux frontières	Régularisé
Résolution spatiale (N_L)	$N_L = l_y = \{16, 32, 64, 128\}$
Résolution temporelle (N_t)	$N_t = \frac{N_L^2}{\text{Re} \nu}$

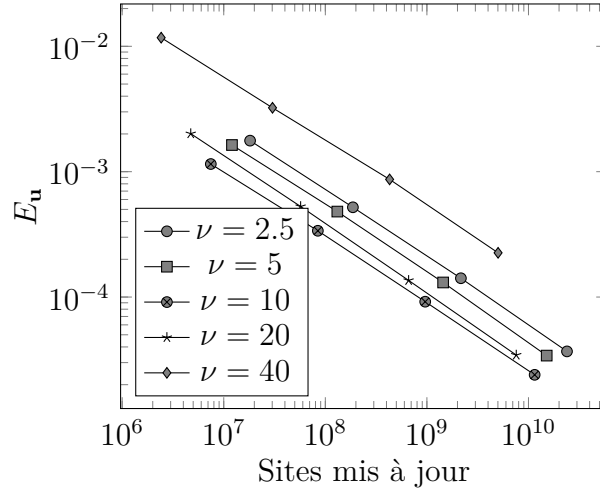
(a) Paramètres de simulation pour la section 4.1.4.



(b) Re = 10.



(c) Re = 1.



(d) Re = 0.1.

Figure 4.5 Erreur E_u en fonction du nombre de sites mis à jour, pour différents nombres de Reynolds et viscosités

4.2 Cavit  carr e 2D avec paroi entra n e

Pour ce cas test, le domaine  tudi  est un carr  en 2 dimensions. Le domaine de simulation est illustr    la figure 4.6. La paroi sup rieure se d place   vitesse constante alors que les autres parois sont fixes. Les conditions initiales sont d finies ainsi :

$$\mathbf{u}|_{t=0} = \mathbf{0} \quad (4.15)$$

$$\rho|_{t=0} = 1 \quad (4.16)$$

Les conditions aux fronti res sont d finies ainsi :

$$\mathbf{u}|_{x=0} = \mathbf{u}|_{x=l_x} = \mathbf{0} = \mathbf{u}|_{y=0} = \mathbf{0} \quad (4.17)$$

$$\mathbf{u}|_{y=l_y} = U_0 \mathbf{i} + 0 \mathbf{j} \quad (4.18)$$

Le nombre de Reynolds est calcul    partir de la vitesse de la paroi sup rieure qui correspond   vitesse caract ristique U_0 . La longueur caract ristique est la largeur du carr  l :

$$\text{Re} = \frac{U_0 l}{\nu} \quad (4.19)$$

Les simulations sont effectu es pour des nombres de Reynolds de 100, 1000 et 7500 pour lesquels la solution en r gime permanent demeure stationnaire. Le crit re d'arr t des simulations est fix  lorsque la dur e de la simulation atteint $\tilde{t} = 200$ afin d'atteindre le r gime permanent pour chaque nombre de Reynolds consid r . Le temps \tilde{t} correspond au temps adimensionnel, soit $\tilde{t} = t/N_t$. La cavit  est un syst me ferm  en terme de transfert massique, cependant les simulations effectu es   l'aide de la m thode LB permettent d'observer que la masse n'est pas conserv e. Ceci est caus  par le type de conditions aux fronti res utilis . Afin de mesurer la quantit  de masse perdue par le syst me, on approxime le flux massique adimensionnelle \tilde{m} sortant du domaine au pas de temps n :

$$\tilde{m}_n = \frac{\partial \tilde{m}}{\partial t} \cong \frac{\tilde{m}_{n-1} - \tilde{m}_n}{\tilde{\delta}_t} \quad (4.20)$$

O  \tilde{m}_n est la masse de fluide adimensionnel   l'int rieur de la cavit  au pas de temps n . La masse de fluide m_n , en unit s LBM, dans la cavit  au pas de temps n est calcul e ainsi :

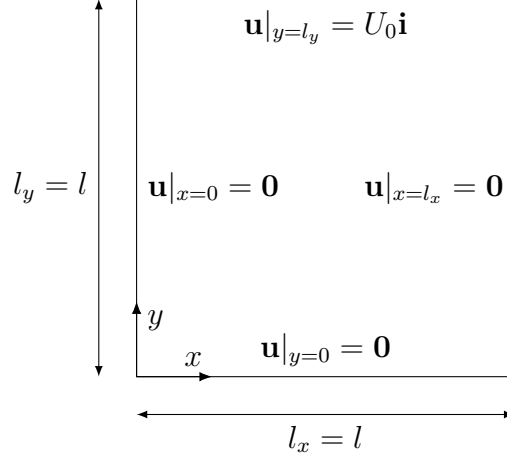


Figure 4.6 Schéma du domaine de calcul pour la cavité 2D

$$m_n = \sum_i \rho_{i,n} \delta_x^2 = \sum_i \rho_{i,n} \quad (4.21)$$

Où $\rho_{i,n}$ correspond à la masse volumique de la cellule i au pas de temps n et δ_x^2 est égale au volume d'une cellule. La masse caractéristique m_0 de l'écoulement est la masse totale de fluide pour $\rho = \rho_0$:

$$m_0 = \sum_i \rho_0 \delta_x^2 = N_L^2 \rho_0 \delta_x^2 = N_L^2 \quad (4.22)$$

Où N_L^2 correspond au nombre de différentes cellules i . La masse adimensionnelle \tilde{m}_n s'exprime donc ainsi :

$$\tilde{m}_n = \frac{m_n}{m_0} = \frac{\sum_i \rho_{i,n}}{N_L^2} \quad (4.23)$$

Elle correspond en quelque sorte à la masse volumique moyenne $\bar{\rho}$ au pas de temps n :

$$\bar{\rho} = \frac{\sum_i \rho_{i,n}}{N_L^2} \quad (4.24)$$

4.2.1 Effet de la résolution sur la perte de masse

Les paramètres de simulation utilisés sont résumés au tableau 4.7a. Dans la figure 4.7b, on observe la dépendance de la perte de masse pour différentes résolutions. On constate

Paramètre	
Nombre de Reynolds (Re)	1000
Modèle(s) de collision	BGK
Modèle(s) de condition aux frontières	Régularisé
Résolution spatiale (N_L)	$N_L = l = 32, 64, 128, 256, 512$
Résolution temporelle (N_t)	$N_t = \frac{N_L^2}{\text{Re} \nu}$

(a) Paramètres de simulation pour la section 4.2.1.

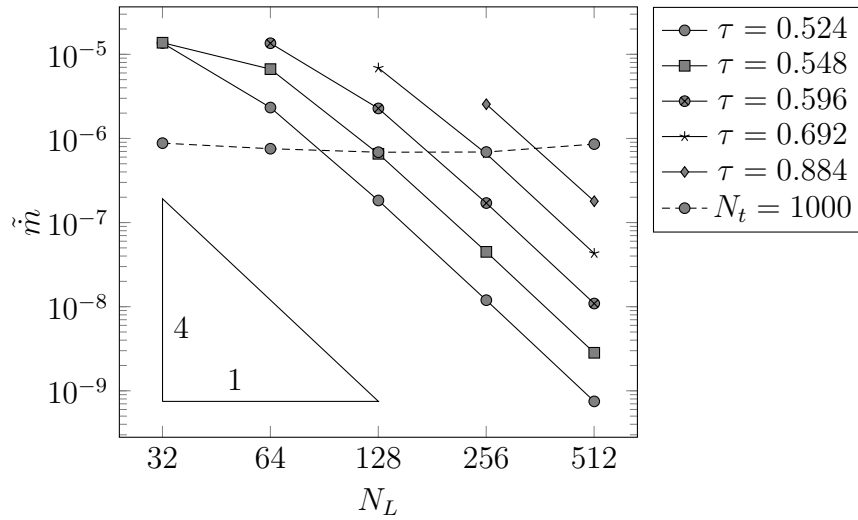
(b) Perte de masse au temps et $\tilde{t} = 200$ en fonction de la résolution N_L pour différents temps de relaxation τ avec superposition d'une courbe à résolution temporelle constante.

Figure 4.7 Réduction de la perte de masse lors du raffinement de maillage

que la perte de masse décroît d'un ordre 4 par rapport au raffinement spatial (N_L). On rappelle cependant que lorsque le τ est maintenu constant, le raffinement temporel (N_t) est quadratique par rapport au raffinement spatial (N_L). Une courbe à résolution temporelle constante $N_t = 1000$ est superposée au graphique 4.7b. Ceci nous permet finalement de conclure que la perte de masse est indépendante de la résolution spatiale et dominée par la discrétisation temporelle. On constate donc une convergence d'ordre 2 de la perte de masse par rapport à la résolution temporelle. Ceci explique pourquoi on observe une convergence d'ordre 4 à la figure 4.7b lorsque $N_t \propto N_L^2$ et une convergence d'ordre 2 à la figure 4.9b lorsque $N_t \propto U_0$.

L'influence des prochains paramètres numériques présentés sur la perte de masse est donc étudiée en fonction de la résolution temporelle. Pour ces cas-ci, la résolution temporelle est contrôlée par la vitesse caractéristique U_0 .

4.2.2 Effet du modèle de condition aux frontières sur la perte de masse

Les paramètres de simulation utilisés sont résumés au tableau 4.8a où la résolution temporelle est fixée par le choix de la vitesse caractéristique U_0 . Dans la figure 4.8b, on observe la perte de masse en fonction de la vitesse caractéristique U_0 pour 4 conditions aux frontières. On voit que les 4 conditions aux frontières étudiées perdent de la masse. On constate que la perte de masse diminue lorsque la résolution temporelle augmente. On conclut alors que la perte de masse deviendrait nulle dans le cas limite. Pour les simulations à plus haute résolution spatiale ($U_0 < 10^{-3}$), seules les conditions Skordos et Régularisé étaient assez stables pour permettre d'effectuer les simulations, contrairement aux conditions d'Inamuro et Zou-He.

4.2.3 Effet de l'opérateur de collision sur la perte de masse

Les paramètres de simulation utilisés sont résumés au tableau 4.9a. Dans la figure 4.9b on observe l'influence du modèle utilisé sur la perte de masse en fonction du temps. On remarque une faible dépendance de l'opérateur de collision à la perte de masse, ce qui est cohérent avec le fait que ce soit les conditions aux frontières qui sont à l'origine de la perte de masse.

4.2.4 Effet de la résolution sur le champ de pression

Les paramètres de simulation utilisés sont résumés au tableau 4.1, où la résolution temporelle est fixée en fonction de U_0 . Les champs de pressions sont évalués à partir de la pression manométrique adimensionnelle \tilde{p} :

Paramètre

Nombre de Reynolds (Re)	1000
Modèle(s) de collision	BGK
Modèle(s) de condition aux frontières	Régularisé, Skordos, Zou-He, Inamuro
Résolution spatiale (N_L)	$N_L = l = 256$
Résolution temporelle (N_t)	$N_t = \frac{N_L}{U_0} = \frac{256}{U_0}$

(a) Paramètres de simulation pour la section 4.2.2.

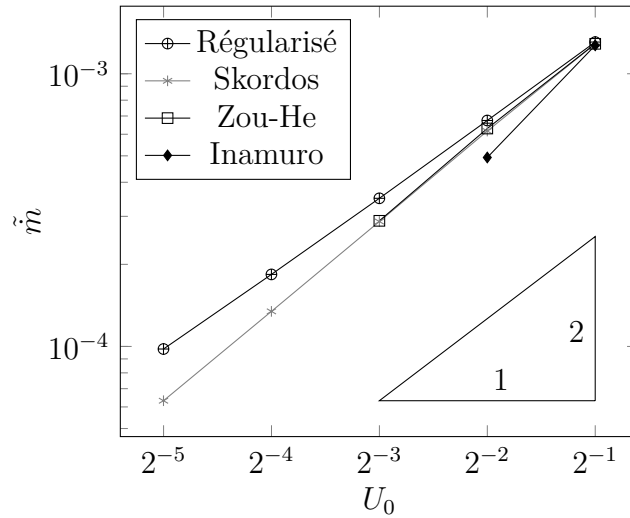
(b) Perte de masse au temps $\tilde{t} = 200$ en fonction de U_0 pour différents modèles de conditions aux frontières

Figure 4.8 Effet du modèle de conditions aux frontières sur la perte de masse lors du raffinement du pas de temps

Tableau 4.1 Paramètres de simulation pour la section 4.2.4

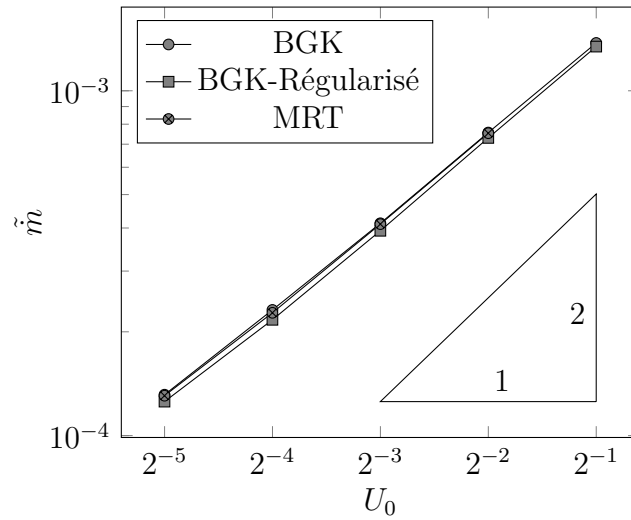
Paramètre

Nombre de Reynolds (Re)	100, 1000, 7500
Modèle(s) de collision	BGK
Modèle(s) de condition aux frontières	Régularisé
Résolution spatiale (N_L)	$N_L = l = 256$
Résolution temporelle (N_t)	$N_t = \frac{N_L}{U_0} = \frac{256}{U_0}$

Paramètre

Nombre de Reynolds (Re)	7500
Modèle(s) de collision	BGK, BGK-Régularisé, MRT
Modèle(s) de condition aux frontières	Régularisé
Résolution spatiale (N_L)	$N_L = l = 256$
Résolution temporelle (N_t)	$N_t = \frac{N_L}{U_0} = \frac{256}{U_0}$

(a) Paramètres de simulation pour la section 4.2.3.



(b) Perte de masse au temps $\tilde{t} = 200$ en fonction de U_0 pour différents opérateurs de collisions.

Figure 4.9 Effet de l'opérateur de collision sur la perte de masse lors du raffinement du pas de temps

$$\tilde{p} = \frac{p}{\bar{\rho}U_0^2} - \tilde{p}_{vortex} = \frac{\rho c_s^2}{\bar{\rho}U_0^2} - \tilde{p}_{vortex} \quad (4.25)$$

Où \tilde{p}_{vortex} correspond à la pression adimensionnelle au centre du vortex pour chaque simulation. La pression est adimensionnalisée par rapport à la masse volumique moyenne $\bar{\rho}$ défini à l'équation (4.24) plutôt que par rapport à ρ_0 afin de compenser pour la perte de masse. Pour le cas où les frontières respecteraient la conservation de masse, la masse volumique moyenne serait toujours égale ρ_0 .

On observe l'effet du pas de temps sur le champ de pression à la figure 4.10. La pression dans le coin supérieur gauche atteint une valeur très basse alors que celle dans le coin supérieur droit atteint une valeur très élevée. Le minimum et le maximum de l'échelle des schémas sont déterminés en faisant abstraction des valeurs extrêmes atteintes aux coins supérieurs. Le centre du vortex central est fixé à une pression manométrique nulle. Dix contours sont tracés à intervalle régulier entre le minimum et le maximum de l'échelle.

On constate une pression en damier pour les simulations où le temps de relaxation τ est très près de la limite de stabilité ($\tau > 0.5$), ce qui correspond à une viscosité nulle. Ceci montre un inconvénient de déterminer la résolution temporelle à partir de la vitesse caractéristique U_0 plutôt qu'à partir de τ directement lorsqu'on fait varier le nombre de Reynolds. Cette instabilité semble être plus intense près des singularités dans les coins supérieurs du domaine. On remarque une différence d'échelle de pression pour les 2 résolutions temporelles différentes, soit $U_0 = 2^{-1}$ et $U_0 = 2^{-5}$. Un pas de temps plus grand augmente l'erreur de compressibilité ainsi que la perte de masse. Ces deux phénomènes peuvent expliquer les différences d'échelle de pression observées pour différents U_0 .

Ayant déjà traité des champs de vitesse et de pression dans les deux derniers cas tests, le prochain cas test est plutôt choisi afin de traiter des contraintes engendrées par l'écoulement. Celui-ci s'attarde donc à la force exercée par l'écoulement sur un obstacle solide.

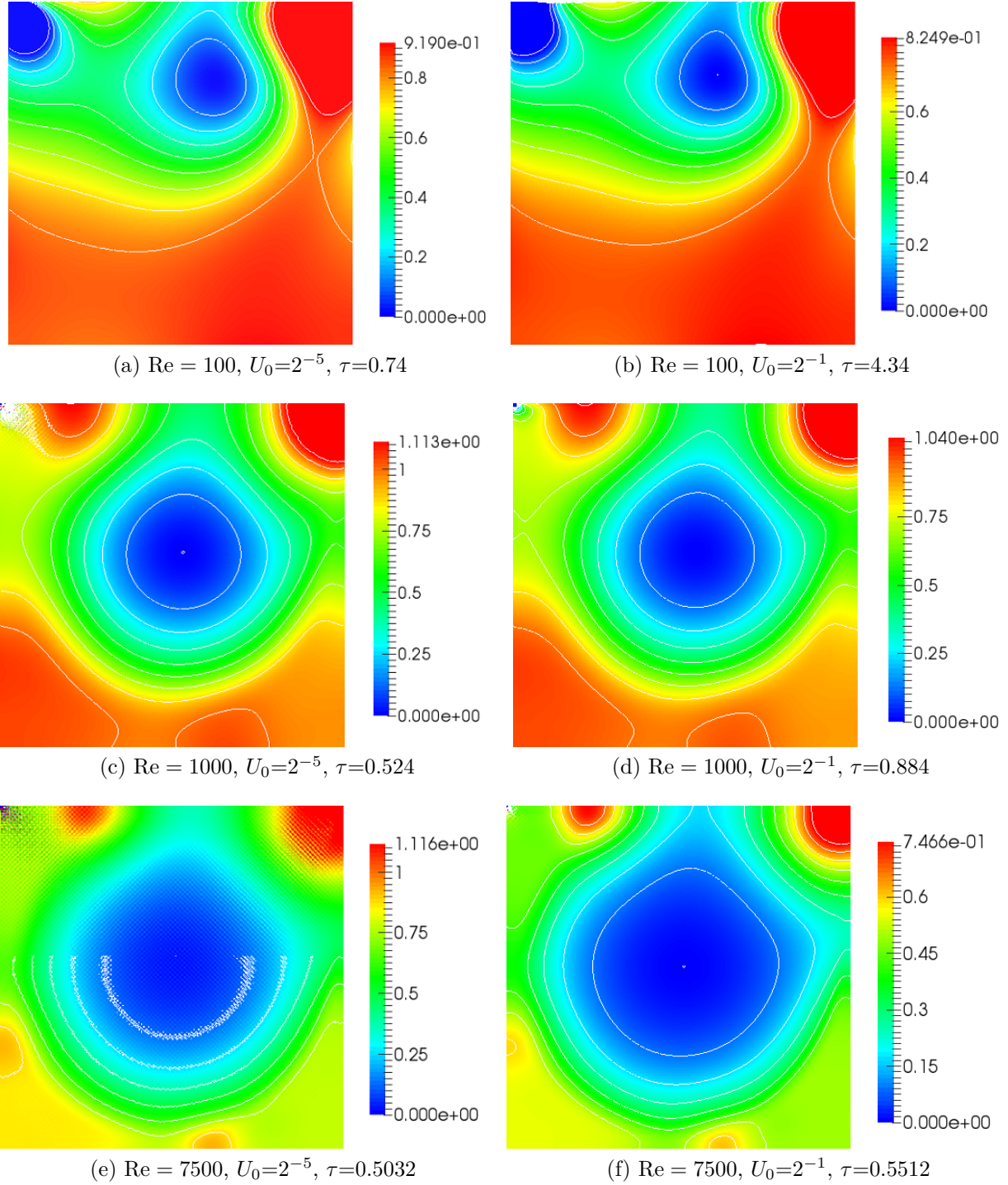


Figure 4.10 Champs de pression adimensionnelle à $\tilde{t} = 200$ pour différents U_0 et nombres de Reynolds

4.3 Écoulement de Poiseuille 2D entre deux plaques avec obstacle cylindrique

Pour le présent cas test, le coefficient de traînée C_d est calculé sur un obstacle cylindrique dans un écoulement de Poiseuille 2D entre deux plaques. Le C_d est calculé pour différents nombres de Reynolds et différentes tailles de l'obstacle. Une analyse de sensibilité est tout d'abord effectuée sur les principaux paramètres numériques. Cette analyse permet de déterminer les paramètres numériques utilisés pour générer les résultats finaux. Ceux-ci sont ensuite comparés avec une référence de la littérature [35]. Cette référence emploie la méthode des éléments finis du logiciel commercial FLUENT 6.2.

Le domaine de calcul est illustré à la figure 4.11. La vitesse est imposée à l'entrée et la pression est imposée à la sortie. Les conditions initiales sont établies à partir des équations analytiques pour un écoulement de Poiseuille 2D incompressible sans obstacle :

$$\mathbf{u}|_{t=0} = U_{max} 4y(l_y - y) \mathbf{i} + 0\mathbf{j} \quad (4.26)$$

$$\rho|_{t=0} = 1 + \frac{8\nu\rho U}{l_y^2 c_s^2} \left(\frac{l_x}{2} - x \right) \quad (4.27)$$

Le champ de vitesse analytique est maintenu à l'entrée alors qu'une pression constante est maintenue à la sortie :

$$\mathbf{u}|_{x=0} = U_{max} 4y(l_y - y) \mathbf{i} + 0\mathbf{j} \quad (4.28)$$

$$\rho|_{x=l_x} = \frac{p|_{x=l_x}}{c_s^2} = 1 \quad (4.29)$$

Il aurait été préférable que la masse volumique unitaire de la condition initiale soit fixée à $x = l_x$ plutôt qu'à $x = l_x/2$ afin d'être cohérente avec la condition de frontière imposée à la sortie. Le modèle de conditions aux frontières Régularisé est utilisé pour toutes les parois du domaine hormis la paroi du cylindre, pour laquelle la condition *bounce-back* est utilisée. Ce choix est dû au fait que l'implémentation de parois *bounce-back* est beaucoup plus simple pour des surfaces courbes. La condition de non-glissement est imposée sur les deux plaques à partir du modèle régularisé :

$$\mathbf{u}|_{y=0} = \mathbf{0} \quad (4.30)$$

$$\mathbf{u}|_{y=l_y} = \mathbf{0} \quad (4.31)$$

La condition de non-glissement $\mathbf{u} = \mathbf{0}$ sur le cylindre est imposée à partir du modèle *bounce-*

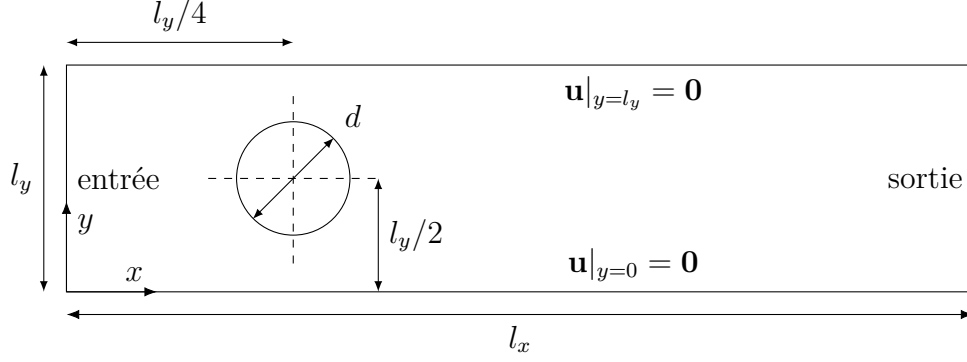


Figure 4.11 Schéma du domaine de calcul pour l'écoulement de Poiseuille avec obstacle cylindrique

back. La région occupée par le cylindre est définie ainsi :

$$\left(x - \frac{l_x}{5}\right)^2 + \left(y - \frac{l_y}{2}\right)^2 \leq \left(\frac{d}{2}\right)^2 \quad (4.32)$$

Où d est le diamètre du cylindre. Le coefficient de traînée C_d sur l'obstacle cylindrique est évalué ainsi :

$$C_d = \frac{F_D}{\frac{1}{2}\rho U_{max}^2 d} \quad (4.33)$$

Où F_D est la force de traînée exercée sur l'obstacle. Afin d'être comparable avec la référence [35], le nombre de Reynolds est calculé à partir de la vitesse moyenne U_{moy} du profil imposée à l'entrée, et du diamètre de l'obstacle cylindrique :

$$\text{Re} = \frac{U_{moy} d}{\nu} \quad (4.34)$$

Pour un écoulement de Poiseuille, la vitesse moyenne U_{moy} est proportionnelle à la vitesse maximale U_{max} :

$$U_{max} = \frac{3}{2} U_{moy} \quad (4.35)$$

Le ratio de blocage β de l'écoulement est défini ainsi :

$$\beta = l_y/d \quad (4.36)$$

Celui-ci permet de contrôler le diamètre du cylindre d par rapport à la distance entre les plaques l_y . La dépendance du C_d par rapport à plusieurs paramètres numériques est étudiée.

On s'assure tout d'abord que le C_d calculé est suffisamment insensible aux paramètres numériques choisis. Cette étude de sensibilité est effectuée pour la résolution spatiale et temporelle ainsi que la longueur l_x du domaine de calcul. Une fois les paramètres numériques choisis, les simulations sont effectuées pour plusieurs nombres de Reynolds entre 1 et 40 et plusieurs ratios de blocage β entre 1.1 et 4.

La force de traînée est calculée numériquement à partir de la méthode d'échange de quantité de mouvement (*momentum exchange method*) [36]. Cette méthode numérique permet d'évaluer la force exercée par le fluide sur une paroi solide. Il s'agit d'effectuer une somme vectorielle de la quantité de mouvement des particules incidentes sur la paroi et des particules réfléchies par celle-ci. Cette somme doit être effectuée à chaque interface entre une cellule de type «solide» et une cellule de type «fluide» pour la totalité de la surface de l'objet considéré.

Le critère d'arrêt de la simulation est établi en observant les fluctuations de la force de traînée en fonction du temps. À chaque 1000 itérations, on retient la valeur de la force de traînée. À chaque 30000 itérations, on évalue la moyenne et l'écart type des 30 dernières forces de traînées retenues. Le critère d'arrêt est établi sur le coefficient de variation c_v , c'est-à-dire l'écart type relatif : $c_v = \sigma/\mu$. La solution est considérée comme suffisamment près du régime stationnaire pour une valeur de $c_v < 10^{-5}$. On ne considère alors que 5 chiffres significatifs sur le coefficient de traînée obtenu. Lors de l'étude de la sensibilité aux paramètres numériques, l'erreur relative [37] est approximée ainsi :

$$e_a = \left| \frac{C_{d,fin} - C_{d,grossier}}{C_{d,fin}} \right| \quad (4.37)$$

Où $C_{d,fin}$ est le coefficient de traînée calculé sur un maillage fin et $C_{d,grossier}$ est calculé sur un maillage grossier. Pour les présentes simulations, $N_{L,fin}/N_{L,grossier} = 2$. Le comportement irrégulier du coefficient C_d lors du raffinement de maillage ne permet pas d'obtenir un bon résultat en appliquant la procédure d'extrapolation de l'erreur décrite dans la référence [37]. Ce phénomène est dû à la représentation des surfaces courbes par des segments perpendiculaires en forme d'escalier. En effet, la condition de frontière *bounce-back* ne permet pas de représenter des surfaces courbes, celles-ci sont alors représentées par des segments de surface parallèles aux axes x et y . Ceci cause une variation irrégulière de la géométrie représentée numériquement lorsque le maillage est raffiné. La variation irrégulière du coefficient C_d lors de la réduction de la résolution spatiale est attribuée à ce phénomène, qui est approfondi au prochain cas test. Les paramètres numériques employés pour les simulations qui suivent sont résumés au tableau 4.2.

Tableau 4.2 Paramètres de simulation pour les sections 4.3.1, 4.3.2 et 4.3.3

Paramètre	
Nombre de Reynolds (Re)	1, 40
Modèle(s) de collision	BGK
Modèle(s) de condition aux frontières	Régularisé et <i>bounce-back</i>
Résolution spatiale (N_L)	$N_L = l_y = 32, 64, 128, 256$
Résolution temporelle (N_t)	$N_t _{U=2^{-8}} = \frac{N_L}{2^{-8}}$

4.3.1 Effet de la longueur du conduit sur le coefficient de traînée C_d

On observe l'effet de la longueur du conduit l_x sur le C_d dans le tableau 4.3. La valeur du C_d ne varie pas de manière significative par rapport à la longueur du conduit, la valeur $l_x/l_y = 8$ est considérée comme suffisante.

4.3.2 Effet de la résolution temporelle sur le coefficient de traînée C_d

Le tableau 4.4 permet d'observer l'effet de la résolution temporelle sur le C_d . Le pas de temps est contrôlé à partir du paramètre $U_0 = N_L/N_t$. Une valeur de U_0 plus petite entraîne une résolution temporelle plus élevée. On observe qu'il aurait été préférable de raffiner le pas de temps pour des valeurs plus élevées du nombre de Reynolds pour lesquels C_d dépend plus fortement du pas de temps. Le coefficient C_d semble varier de manière régulière et monotone en fonction du raffinement temporel, ce qui porte à croire qu'il serait possible d'extrapoler l'erreur sur le C_d dû à la discrétisation temporelle.

Tableau 4.3 Effet de la longueur du conduit l_x sur le coefficient de traînée C_d pour $\beta = 2$ et $N_L = 256$. $\Delta C_d(l_x/l_y) = |C_d(l_x/l_y) - C_d(l_x/2l_y)|$

l_x/l_y	Re = 1			Re = 40		
	C_d	ΔC_d	$\Delta C_d/C_d$ [%]	C_d	ΔC_d	$\Delta C_d/C_d$ [%]
4	117.55	-	-	4.1812	-	-
8	117.55	$4.97 \cdot 10^{-4}$	$4.23 \cdot 10^{-4}$	4.1902	$9.02 \cdot 10^{-3}$	0.22
12	117.55	$1.17 \cdot 10^{-3}$	$9.93 \cdot 10^{-4}$	4.1939	$3.69 \cdot 10^{-3}$	$8.81 \cdot 10^{-2}$
16	117.55	$1.24 \cdot 10^{-3}$	$1.05 \cdot 10^{-3}$	4.1972	$3.31 \cdot 10^{-3}$	$7.88 \cdot 10^{-2}$

Tableau 4.4 Effet de U_0 sur le coefficient de traînée C_d pour $\beta = 2$ et $N = 256$. $\Delta C_d(N_t) = |C_d(N_t) - C_d(N_t/2)|$

U	Re = 1			Re = 40		
	C_d	ΔC_d	$\Delta C_d/C_d$ [%]	C_d	ΔC_d	$\Delta C_d/C_d$ [%]
2^{-9}	117.86	0.31	0.26	4.1368	$5.34 \cdot 10^{-2}$	1.29
2^{-8}	117.55	0.36	0.31	4.1902	0.31	7.28
2^{-7}	117.19	0.66	0.56	4.4954	1.69	37.5
2^{-6}	116.54	1.87	1.6	2.8097	1.89	67.37
2^{-5}	114.67	5.68	4.96	0.9167	-	-
2^{-4}	108.98	-	-	-	-	-

Tableau 4.5 Effet de la résolution spatiale N_L sur le coefficient de traînée C_d pour $\beta = 2$. $\Delta C_d(N_L) = |C_d(N_L) - C_d(N_L/2)|$

N_L	Re = 1			Re = 40		
	C_d	ΔC_d	$\Delta C_d/C_d$ [%]	C_d	ΔC_d	$\Delta C_d/C_d$ [%]
32	115.8	-	-	-	-	-
64	116.47	0.67	0.58	1.7053	-	-
128	117.36	0.89	0.76	4.4988	2.79	62.1
256	117.55	0.19	0.16	4.1902	0.31	7.37

4.3.3 Effet de la résolution spatiale sur le coefficient de traînée C_d

On observe l'effet de la résolution spatiale N_L dans le tableau 4.5. On rappelle que pour une viscosité ν constante, le pas de temps est également raffiné lorsque le pas en espace est raffiné. La valeur de la viscosité est de $\nu = 1$ pour un $Re = 1$ et de $\nu = 1/40$ pour $Re = 40$. On remarque que, contrairement au raffinement temporel, le raffinement spatial ne mène pas à une variation monotone du C_d . Ceci est dû à la variation de la géométrie représentée numériquement lors du raffinement spatial, tel que déjà discuté. En effet, les parois en escalier de type *bounce-back* de l'obstacle cylindrique induisent une variation irrégulière du volume et de la surface du cylindre en fonction de la résolution. D'autre part, on observe qu'il serait nécessaire d'utiliser une résolution plus fine lorsque le nombre de Reynolds est plus élevé. On s'attend alors à obtenir de moins bonnes approximations pour des nombres de Reynolds plus élevés. L'erreur relative approximative $\frac{\Delta C_d}{C_d}$ permet de considérer environ 4 chiffres significatifs à bas nombre de Reynolds et 2 chiffres significatifs à haut nombre de Reynolds.

4.3.4 Comparaison des coefficients de traînée C_d obtenus

Les paramètres numériques employés pour les simulations finales sont résumés au tableau 4.6. La longueur du domaine employée est de $l_x/l_y = 8$. Les résultats obtenus sont comparés avec la référence [35] au tableau 4.7. On remarque que les résultats coïncident assez bien avec ceux de la référence. Les résultats diffèrent beaucoup pour le cas avec $\beta = 1.1$ puisque le nombre de noeuds entre l'obstacle et les parois est très petit, soit près de 4 noeuds de largeur. L'espace entre la paroi et l'obstacle est pourtant un endroit critique de l'écoulement où les gradients de vitesses sont élevés. Ceci démontre bien l'inconvénient d'un maillage cartésien à pas de temps régulier, il aurait été avantageux de raffiner le maillage entre les parois et le cylindre, surtout quand ce passage est étroit. L'effet du maillage cartésien sur la précision est approfondi dans le cadre des deux prochains cas tests.

Les trois premiers cas tests traitent d'écoulements 2D et ne considèrent pas l'application

Tableau 4.6 Paramètres de simulation pour la section 4.3.4

Paramètre	
Nombre de Reynolds (Re)	1, 5, 10, 20, 30, 40
Modèle(s) de collision	BGK
Modèle(s) de condition aux frontières	Régularisé
Résolution spatiale (N_L)	$N_L = l_y = 256$
Résolution temporelle (N_t)	$N_t _{U=2^{-8}} = \frac{N_L}{U} = 2^{16}$

Tableau 4.7 Comparaison des coefficients de traînée C_d pour différents β et Re

Re	C_d présent travail	C_d référence [35]	Différence relative [%]
$\beta = 4$			
1	28.5046	28.5357	0.109
5	6.2644	6.2700	0.090
10	3.6729	3.6733	0.011
20	2.3992	2.3904	0.369
30	1.9619	1.9413	1.059
40	1.7395	1.7034	2.118
$\beta = 2$			
1	117.5519	117.6131	0.052
5	23.7409	23.7471	0.026
10	12.1962	12.1905	0.047
20	6.6661	6.6411	0.376
30	4.9746	4.9258	0.992
40	4.1902	4.1121	1.900
$\beta = 1.6$			
1	281.6384	282.0961	0.162
5	56.5449	56.6210	0.134
10	28.5967	28.6119	0.053
20	14.9258	14.8796	0.311
30	10.6424	10.5331	1.038
40	8.6881	8.5001	2.212
$\beta = 1.1$			
1	15 132.5159	13 108.5217	15.440
5	3 034.0055	2 622.3818	15.697
10	1 530.4455	1 312.2420	16.628
20	806.8569	658.2516	22.576
30	546.2208	441.3092	23.773
40	194.7114	333.7845	41.666

finale du calcul de la perméabilité de milieux poreux. Les deux prochains et derniers cas tests pour leur part sont des écoulements 3D pour lesquels la perméabilité est calculée.

4.4 Écoulement de Poiseuille 3D à travers un cylindre

Pour le présent cas test, la perméabilité d'un pore à faible résolution est calculée afin d'y évaluer la principale source d'erreur. La frontière de type *bounce-back* est employée ; celle-ci peut facilement représenter les frontières courbes par des frontières en escalier. Afin de déterminer la principale source d'erreur dans le calcul de la perméabilité, on distingue trois types d'erreurs : la première étant due à la discrétisation de la géométrie, la seconde étant due à la représentation des surfaces courbes par des parois en escalier, et la troisième correspond à l'erreur intrinsèque de la méthode LB (incluant les modèles de conditions aux frontières). Les paramètres de simulations utilisés pour ce cas test sont résumés au tableau 4.8. La résolution spatiale pour un maillage donné est déterminée par le paramètre D_{eq} défini ci-après, et la résolution temporelle est déterminée à partir du temps de relaxation.

La figure 4.12 illustre la section d'un cylindre discrétisé. La discrétisation est effectuée en superposant un maillage cartésien 3D à pas d'espace constant. Les noeuds sont représentés par des points noirs au centre de cellules cubiques. Le maillage est généré en vérifiant si chaque noeud se situe à l'intérieur ou à l'extérieur du cylindre. L'état solide est attribué aux noeuds se trouvant à l'extérieur du cylindre ; ceux-ci se comportent comme une frontière sans glissement de type *bounce-back*. L'état fluide est attribué aux noeuds se trouvant à l'intérieur du cylindre ; ceux-ci correspondent à des noeuds de discrétisation de l'algorithme LB. Sur la figure 4.12, les cercles en lignes pointillées représentent les diamètres maximum D_{max} et minimum D_{min} d'un cylindre permettant d'obtenir le même maillage, c'est-à-dire une même géométrie discrétisée. L'axe de révolution du cylindre est parallèle à l'axe des x et coïncide avec le noeud d'une cellule. Le cercle en ligne pleine correspond au cercle de diamètre d'aire moyenne D_{moy} . Le diamètre d'aire moyenne D_{moy} est le diamètre du cercle dont l'aire est la moyenne entre l'aire du cercle de diamètre minimum et l'aire du cercle de diamètre maximum.

Tableau 4.8 Paramètres de simulation pour la section 4.4

Paramètre	
Nombre de Reynolds (Re)	0.01
Modèle(s) de collision	BGK, MRT
Modèle(s) de condition aux frontières	Régularisé, <i>bounce-back</i>
Résolution spatiale (N_L)	$N_L = D_{eq} = [4, 32]$
Résolution temporelle (N_t)	$N_t _{\tau=\{0.6, 0.8\}} = \frac{N_L^2}{Re\nu} = \frac{3 N_L^2}{0.01(\tau-0.5)}$

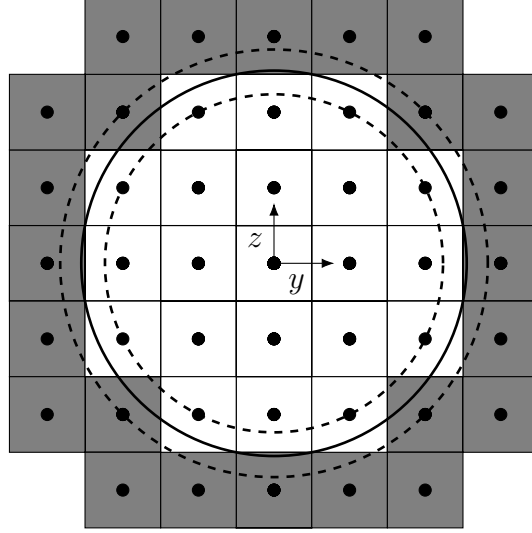


Figure 4.12 Section du domaine de simulation avec représentation d'un des maillages utilisés (différents diamètres menant aux mêmes maillages sont illustrés)

La valeur de D_{moy} est obtenue ainsi :

$$D_{moy} = \left(\frac{D_{min}^2 + D_{max}^2}{2} \right)^{1/2} \quad (4.38)$$

Le diamètre moyen est calculé à partir du carré des diamètres extrêmes afin que l'erreur de discrétisation E_{disc} introduite plus loin soit la même, peu importe si elle est calculée par rapport au diamètre minimum ou maximum. Autrement dit, le diamètre moyen est le diamètre associé à la moyenne des perméabilités des diamètres maximum et minimum. Le diamètre moyen est utilisé afin de décrire la résolution d'un maillage donné en regroupant une plage de résolutions menant au même maillage.

Le diamètre équivalent D_{eq} est le diamètre d'un cercle de même aire que la section du cylindre discrétisé :

$$D_{eq} = \left(\frac{4 A_{num}}{\pi} \right)^{1/2} \quad (4.39)$$

Par exemple, l'aire de la section de cylindre discrétisé de la figure 4.12 est de $A_{num} = 21$. Le nombre de Reynolds est calculé à partir des vitesses et longueurs caractéristiques $U_0 = U_{max}$ et $L_0 = D_{eq}$:

$$Re = \frac{U_{max} D_{eq}}{\nu} \quad (4.40)$$

Où U_{max} est la vitesse maximale du profil de vitesse de la solution analytique de Poiseuille. Le nombre de Reynolds est fixé à $Re = 0.01$ afin de s'assurer que l'écoulement soit de type

rampant, dominé par les forces visqueuses. Ce choix est fait afin d'obtenir un nombre de Darcy unique pour une géométrie poreuse donnée. En effet, tel que vu à la section (3.2.5), la loi de Darcy est déduite des équations de Navier-Stokes et de la loi de Stokes pour un écoulement rampant. Pour un écoulement non-rampant, on ne peut supposer que la perméabilité est constante et elle dépendrait du régime de l'écoulement (nombre de Reynolds).

Les conditions initiales sont déterminées à partir des solutions analytiques de l'équation de Poiseuille à travers un cylindre [23] :

$$\mathbf{u}|_{t=0} = U_0 \left(1 - \frac{4}{D_{eq}^2} (y^2 + z^2) \right) \quad (4.41)$$

$$\rho|_{t=0} = \frac{p|_{t=0}}{c_s^2} = 1 + \frac{16 \nu \rho U_{max}}{D_{eq}^2 c_s^2} \left(\frac{l}{2} - x \right) \quad (4.42)$$

Où l correspond à la longueur du cylindre. Pour les présentes simulations, $l = 8 D_{eq}$. Les conditions aux frontières régularisées sont ensuite employées afin d'imposer une différence de pression entre l'entrée et la sortie du cylindre :

$$\rho|_{x=0} = \frac{p|_{x=0}}{c_s^2} = 1 + \frac{16 \nu \rho U_{max}}{D_{eq}^2 c_s^2} \frac{l}{2} \quad (4.43)$$

$$\rho|_{x=l} = \frac{p|_{x=l}}{c_s^2} = 1 - \frac{16 \nu \rho U_{max}}{D_{eq}^2 c_s^2} \frac{l}{2} \quad (4.44)$$

Les conditions aux frontières de type *bounce-back* sont employées afin de représenter les parois sans glissement du cylindre. Le cylindre peut être décrit par l'équation suivante :

$$y^2 + z^2 \geq \left(\frac{D_{eq}}{2} \right)^2 \quad (4.45)$$

La figure 4.12 illustre la section du cylindre alors que la figure 4.13 illustre sa longueur.

4.4.1 Analyse des sources d'erreurs sur la perméabilité calculée

La perméabilité est calculée numériquement à partir de la loi de Darcy de l'équation (3.58). On ne considère que l'écoulement à l'intérieur du cylindre dont la porosité est unitaire $\phi = 1$. Il n'y a pas de force externe, donc $\mathbf{g} = \mathbf{0}$, et l'on ne considère que la composante en x puisque le gradient de pression imposée est nulle dans les deux autres directions. On obtient donc la loi de Darcy en une seule dimension et sans force externe :

$$\bar{u}_x = -\frac{\kappa}{\mu} \frac{\partial p}{\partial x} \quad (4.46)$$

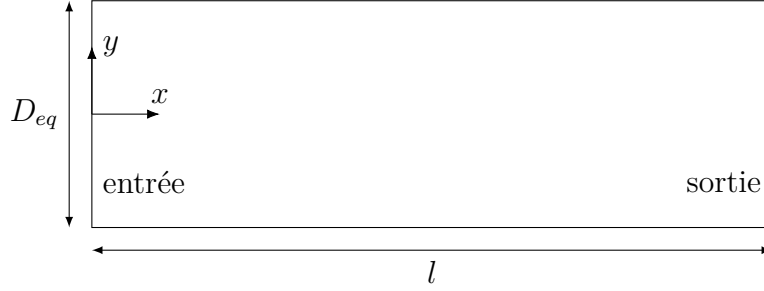


Figure 4.13 Schéma du domaine de calcul pour l'écoulement de Poiseuille en 3D à travers un cylindre

Le gradient de pression en x est approximé par la différence de pression imposée aux extrémités du cylindre. Cette différence de pression est elle-même imposée à partir de l'équation analytique de Poiseuille 3D :

$$\frac{\partial p}{\partial x} \cong \frac{\Delta p}{\Delta x} = \frac{16 \mu U_{max}}{D^2} \quad (4.47)$$

La perméabilité numérique κ_n est calculée numériquement à partir des simulations. Afin de la calculer, l'équation (4.47) est substituée dans l'équation (4.46) et le diamètre D_{moy} est introduit :

$$\kappa_n = \bar{u}_x \mu \frac{\Delta x}{\Delta p} = \frac{\bar{u}_x D_{moy}^2}{U_0 16} \quad (4.48)$$

La vitesse moyenne \bar{u}_x est évaluée numériquement en calculant la moyenne de la vitesse à l'entrée et à la sortie du cylindre, ce qui permet ensuite d'évaluer κ_n .

Le critère d'arrêt est établi en observant les fluctuations en fonction du temps de la perméabilité évaluée numériquement. La méthode employée est la même que celle décrite à la section 4.3 pour la force de traînée.

La perméabilité théorique κ_t pour sa part est évaluée à partir et de l'équation de Poiseuille 3D, soit :

$$\bar{u}_x = \frac{R^2}{8\mu} \frac{\partial p}{\partial x} \quad (4.49)$$

L'équation (4.49) est substituée dans l'équation (4.46) afin d'exprimer κ_t ainsi :

$$\kappa_t = \frac{R^2}{8} = \frac{D^2}{32} \quad (4.50)$$

Comme illustrés à la figure 4.12, plusieurs cylindres de diamètres différents permettent d'obtenir un même maillage discrétisé. Pour un maillage donné, la perméabilité théorique de l'équation (4.50) dépend du diamètre D de référence. Trois perméabilités théoriques diffé-

rentes sont définies pour un maillage donné, soit celle pour un diamètre minimum, maximum, et moyen :

$$\kappa_{t,min} = \frac{D_{min}^2}{32} \quad (4.51)$$

$$\kappa_{t,max} = \frac{D_{max}^2}{32} \quad (4.52)$$

$$\kappa_{t,moy} = \frac{D_{moy}^2}{32} = \frac{\kappa_{t,min} + \kappa_{t,max}}{2} \quad (4.53)$$

Pour les présentes simulations, différents types d'erreurs ont été définis par l'auteur afin d'évaluer quelle est la principale source d'erreur.

Erreur numérique E_n

Pour un maillage donné, l'erreur numérique E_n est évaluée entre la perméabilité numérique κ_n et la perméabilité théorique $\kappa_{t,moy}$.

$$E_n = \frac{|\kappa_n - \kappa_{t,moy}|}{\kappa_{t,moy}} \quad (4.54)$$

Erreur numérique maximale $E_{n,max}$

Pour un maillage donné, l'erreur numérique maximale $E_{n,max}$ est calculée à partir des perméabilités maximum $\kappa_{t,max}$ et minimum $\kappa_{t,min}$. Ceci permet d'évaluer l'erreur numérique maximale en considérant que différentes géométries réelles mènent à une même géométrie discrétisée.

$$E_{n,max} = \frac{\max(|\kappa_n - \kappa_{t,min}|; |\kappa_n - \kappa_{t,max}|)}{\kappa_{t,moy}} \quad (4.55)$$

Erreur de discrétisation E_{disc}

L'erreur de discrétisation E_{disc} correspond à la portion de l'erreur numérique maximale $E_{n,max}$ due à la discrétisation du milieu. Celle-ci dépend de la méthode de discrétisation employée afin de générer le maillage.

$$E_{disc} = \frac{|\kappa_{t,moy} - \kappa_{t,max}|}{\kappa_{t,moy}} = \frac{|\kappa_{t,moy} - \kappa_{t,min}|}{\kappa_{t,moy}} = \frac{D_{moy}^2 - D_{min}^2}{D_{moy}^2} \quad (4.56)$$

Il est possible d'exprimer $E_{n,max}$ en fonction de E_{disc} et E_n à partir d'une inégalité triangu-

laire :

$$E_{n,max} = \frac{\max(|\kappa_n - \kappa_{t,moy} + \kappa_{t,moy} - \kappa_{t,min}|; |\kappa_n - \kappa_{t,moy} + \kappa_{t,moy} - \kappa_{t,max}|)}{\kappa_{t,moy}} \quad (4.57)$$

$$E_{n,max} \leq \frac{\max(|\kappa_n - \kappa_{t,moy}| + |\kappa_{t,moy} - \kappa_{t,min}|; |\kappa_n - \kappa_{t,moy}| + |\kappa_{t,moy} - \kappa_{t,max}|)}{\kappa_{t,moy}} \quad (4.58)$$

$$E_{n,max} \leq \frac{|\kappa_n - \kappa_{t,moy}| + \max(|\kappa_{t,moy} - \kappa_{t,min}|; |\kappa_{t,moy} - \kappa_{t,max}|)}{\kappa_{t,moy}} \quad (4.59)$$

$$E_{n,max} \leq \frac{|\kappa_n - \kappa_{t,moy}| + |\kappa_{t,moy} - \kappa_{t,min}|}{\kappa_{t,moy}} \quad (4.60)$$

$$E_{n,max} \leq E_{disc} + E_n \quad (4.61)$$

Erreur de maillage E_{maill}

L'erreur de maillage E_{maill} est l'erreur attribuable au maillage généré. Puisque le maillage approxime la géométrie réelle par une géométrie formée de cubes identiques, les frontières sont représentées par des surfaces en escaliers plutôt que par des surfaces courbes. Cette erreur est approximée en évaluant la perméabilité numérique raffinée $\kappa_{n,raff}$. La perméabilité $\kappa_{n,raff}$ est calculée en raffinant un maillage préexistant plutôt qu'en remaillant la géométrie réelle de manière plus raffinée. Autrement dit, chaque cellule est subdivisée en 1^3 , 2^3 ou 3^3 cellules pour un niveau de raffinement de 1, 2 ou 3 respectivement. Ceci permet de garder l'aspect grossier des surfaces en escalier tout en ayant beaucoup plus de noeuds de discrétisation. Pour les présentes simulations, un raffinement de 3 est choisi, donc chaque cellule est subdivisée en 9. Ce raffinement est considéré comme suffisant puisque la différence relative entre $\kappa_{n,raff}$ pour un raffinement de 2 et 3 est toujours inférieur à 1% pour les simulations avec $\tau = 0.6$ et les opérateurs de collision BGK et MRT. L'erreur de maillage est donc définie comme étant la différence relative entre la perméabilité théorique et la perméabilité théorique d'une géométrie avec parois en escalier :

$$E_{maill} = \frac{|\kappa_{n,raff} - \kappa_{t,moy}|}{\kappa_{t,moy}} \quad (4.62)$$

Erreur de la méthode LB E_{LB}

L'erreur de la méthode LB E_{LB} est l'erreur intrinsèque aux modèles de la méthode LB. Celle-ci est calculée en considérant un maillage représentant parfaitement la géométrie réelle. La perméabilité raffinée $\kappa_{n,raff}$ est considérée comme une bonne approximation de la perméabi-

lité théorique pour un cylindre dont la géométrie réelle est formée de parois en escaliers. Cette erreur est approximée en évaluant la différence relative entre la perméabilité d'un maillage raffinée $\kappa_{n,raff}$ et la perméabilité numérique κ_n .

$$E_{LB} = \frac{|\kappa_n - \kappa_{n,raff}|}{\kappa_{t,moy}} \quad (4.63)$$

L'erreur numérique E_n peut également être exprimée en fonction de E_{maill} et E_{LB} à partir d'une inégalité triangulaire :

$$E_n = \frac{|\kappa_n - \kappa_{t,moy}|}{\kappa_{t,moy}} = \frac{|\kappa_n - \kappa_{n,raff} + \kappa_{n,raff} - \kappa_{t,moy}|}{\kappa_{t,moy}} \leq \frac{|\kappa_n - \kappa_{n,raff}| + |\kappa_{n,raff} - \kappa_{t,moy}|}{\kappa_{t,moy}} \quad (4.64)$$

$$E_n \leq E_{LB} + E_{maill} \quad (4.65)$$

En substituant l'équation (4.61) dans l'équation (4.65), on retrouve l'inégalité suivante :

$$E_{n,max} \leq E_{LB} + E_{maill} + E_{disc} \quad (4.66)$$

L'équation (4.66) exprime donc comment l'erreur maximale $E_{n,max}$ est subdivisée en 3 sources d'erreur distinctes. Ces erreurs étant attribuées à la discrétisation de la géométrie E_{disc} , le maillage de type cartésien (cellules cubiques identiques) E_{maill} et la méthode LB (incluant les conditions aux frontières) E_{LB} .

Les erreurs E_{maill} et E_{disc} dépendent de la procédure de génération du maillage. Par exemple, il aurait pu être préférable de déterminer le type solide ou fluide d'une cellule en évaluant la proportion de celle-ci occupée par le fluide. Autrement dit, après avoir superposé le maillage cartésien à la géométrie réelle, on attribue le type fluide à la cellule si plus de la moitié de son volume est occupée par le fluide dans la géométrie réelle, on attribue le type solide autrement. Cette procédure nécessiterait cependant de calculer des intégrales de volumes pour plusieurs cellules chevauchant les parois.

4.4.2 Comparaison des types d'erreurs définis

La figure 4.14 exprime l'erreur numérique maximale $E_{n,max}$ en fonction de la résolution. On observe qu'un pore d'une résolution de 10 cellules de diamètre provoque une erreur numérique maximale $E_{n,max}$ d'approximativement 10%. Il serait donc nécessaire que les pores d'un milieu poreux aient un diamètre de l'ordre de 10 cellules afin que leur perméabilité soit bien représentée.

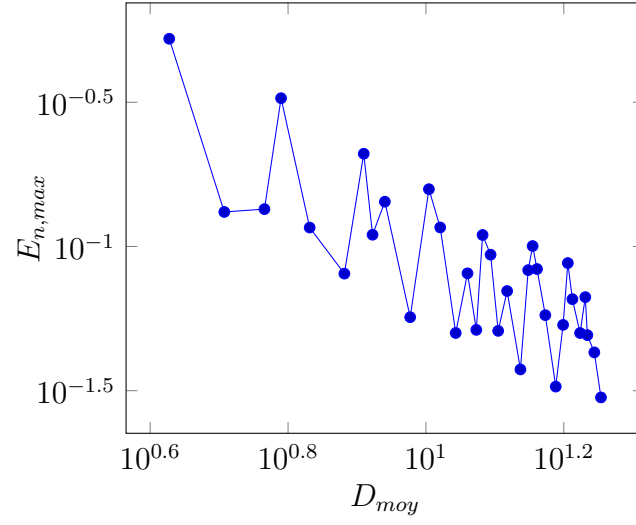


Figure 4.14 Erreur numérique maximale $E_{n,max}$ en fonction de la résolution pour le modèle BGK avec $\tau = 0.6$

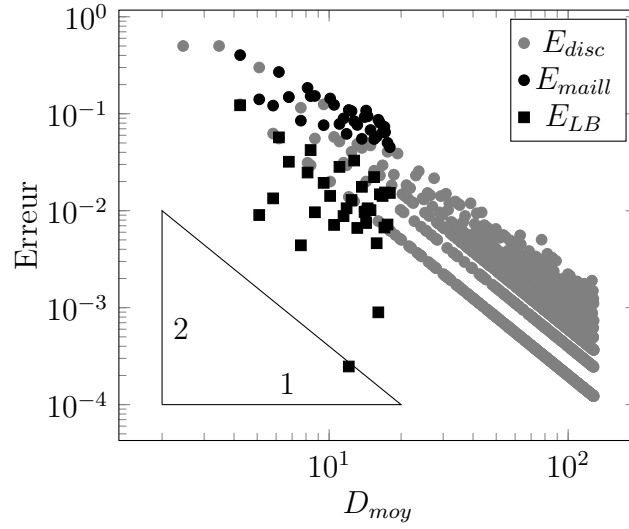


Figure 4.15 Comparaison des différents types d'erreur en fonction de la résolution pour le modèle BGK avec $\tau = 0.6$

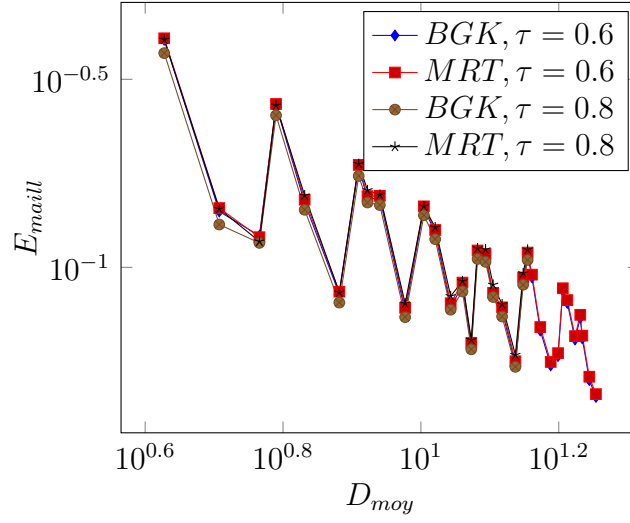


Figure 4.16 Erreur de maillage E_{maill} en fonction de la résolution pour différents opérateurs et temps de relaxation

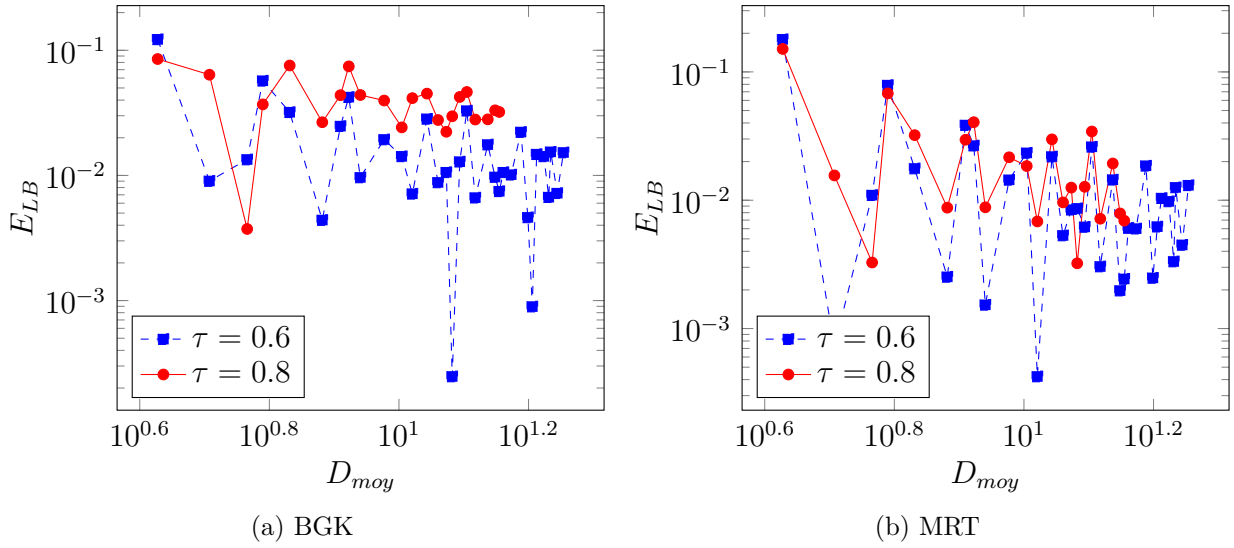


Figure 4.17 Erreur de la méthode LB E_{LB} en fonction de la résolution pour différents opérateurs de collision et temps de relaxation

La figure 4.15 exprime les trois types d'erreurs du membre de droit de l'équation (4.66) pour le modèle BGK avec $\tau = 0.6$. Le but est de comparer leur contribution à l'erreur $E_{n,max}$. L'erreur de maillage E_{maill} , qui est dû à la représentation des surfaces courbes, est prédominante. L'erreur de discrétisation E_{disc} pour sa part dépend uniquement du maillage et non des résultats de simulation. On conclut alors que la principale source d'erreur est de nature géométrique. Celle-ci aurait sûrement été encore plus importante pour un cylindre orienté dans une direction quelconque plutôt que parallèle à l'axe des x puisqu'il y aurait eu présence d'une paroi en escalier dans le sens de l'écoulement.

La figure 4.16 compare l'erreur de maillage pour les différents opérateurs de collision et temps de relaxation. On observe que l'erreur de maillage E_{maill} dépend très peu de l'opérateur de collision et du temps de relaxation, les courbes du graphique de la figure 4.16 sont presque indiscernables. Ceci est cohérent avec la définition de E_{maill} .

La figure 4.17 compare l'erreur E_{LB} de la méthode LB pour les différents opérateurs de collision et temps de relaxation. On observe qu'avec l'opérateur MRT l'erreur E_{LB} dépend peu du temps de relaxation, contrairement à l'opérateur BGK. Pour l'opérateur BGK, le temps de relaxation a un effet significatif sur l'erreur de la méthode E_{LB} . La raison de la différence de sensibilité au temps de relaxation entre le modèle BGK et le modèle MRT est due à la position numérique de la paroi *bounce-back*. La position numérique de la paroi *bounce-back* est dépendante de la viscosité ν (ou τ) [38] ; cette dépendance est moindre pour le modèle BGK que pour le modèle MRT. Les temps de relaxation du modèle MRT peuvent même être optimisés afin de minimiser davantage cette sensibilité [38]. On conclut alors que même l'erreur E_{LB} pourrait être dominée par des effets géométriques, c'est-à-dire la position des parois *bounce-back*.

Finalement, il est plus avantageux de raffiner la résolution spatiale que la résolution temporelle puisque les deux erreurs E_{maill} et E_{disc} sont pratiquement indépendantes de la résolution temporelle. Il serait préférable d'utiliser une résolution temporelle plus grossière (donc augmenter τ) afin de réduire le temps de calcul. En somme, pour une résolution temporelle grossière, le modèle MRT devient avantageux étant donné sa plus grande insensibilité à la résolution temporelle.

Le prochain cas test concerne une géométrie plus près d'une géométrie poreuse. On vérifie la capacité de la méthode LB à calculer la perméabilité de différentes géométries poreuses possédant différentes porosités.

4.5 Écoulement 3D à travers des réseaux de sphères

Le présent cas test concerne un écoulement à travers un réseau compact de sphères. Ce cas test a été choisi dans l'optique de caractériser l'infiltration d'eau dans des sols granulaires. Une telle application implique un écoulement diphasique (air et eau), à travers un milieu poreux, et entraîné par la force gravitationnelle. On entend par écoulement diphasique un écoulement dont une frontière est bien définie entre les phases différentes [39]. Dans ce cas-ci, l'air et l'eau sont considérés comme deux phases immiscibles. Calculer la perméabilité de ces milieux poreux permettrait de mieux prédire leur comportement hydrologique. Pour le présent cas test, on ne considère que le cas simple où le milieu est complètement saturé d'eau, ce qui permet de ne considérer qu'une seule phase liquide. Les granulates sont idéalisés par des réseaux de sphères.

Pour la présente simulation, la perméabilité est calculée en ne considérant qu'un petit élément du milieu poreux. Toutes les frontières du domaine de calcul sont périodiques afin de minimiser la taille du domaine. Trois types de réseaux sont étudiés afin d'obtenir des résultats pour différentes porosités. La figure 4.18 illustre les domaines de simulation pour les trois types de réseaux cubiques de sphères. Le tableau 4.9 résume la porosité de ces réseaux et la largeur l des domaines, où d est le diamètre des sphères. Pour ce cas test, la résolution spatiale $N_L = l$ ne correspond pas toujours à la longueur caractéristique $L_0 = d$.

Les maillages sont générés de la même manière qu'au dernier cas test. C'est-à-dire qu'un maillage cartésien et structuré est superposé aux géométries représentées à la figure 4.18. L'état solide ou fluide est attribué à chaque cellule dépendamment de si le centre de la cellule se trouve à l'extérieur ou à l'intérieur des sphères.

La perméabilité κ , dont les unités sont des m^2 est adimensionnalisée à partir de la longueur caractéristique L_0 , soit le diamètre des sphères $d = L_0$. La perméabilité adimensionnelle est le nombre de Darcy Da :

$$\text{Da} = \frac{\kappa}{L_0^2} = \frac{\kappa}{d^2} \quad (4.67)$$

Tableau 4.9 Largeur et porosité des réseaux cubiques utilisés afin de générer les géométries poreuses

Réseau	Largeur l	Porosité ϕ
Primitif	d	$1 - \pi/6 \approx 0.476$
Centré	$d\sqrt{2}/2$	$1 - \pi\sqrt{3}/8 \approx 0.320$
Faces centrées	$d\sqrt{3}/2$	$1 - \pi\sqrt{2}/6 \approx 0.260$

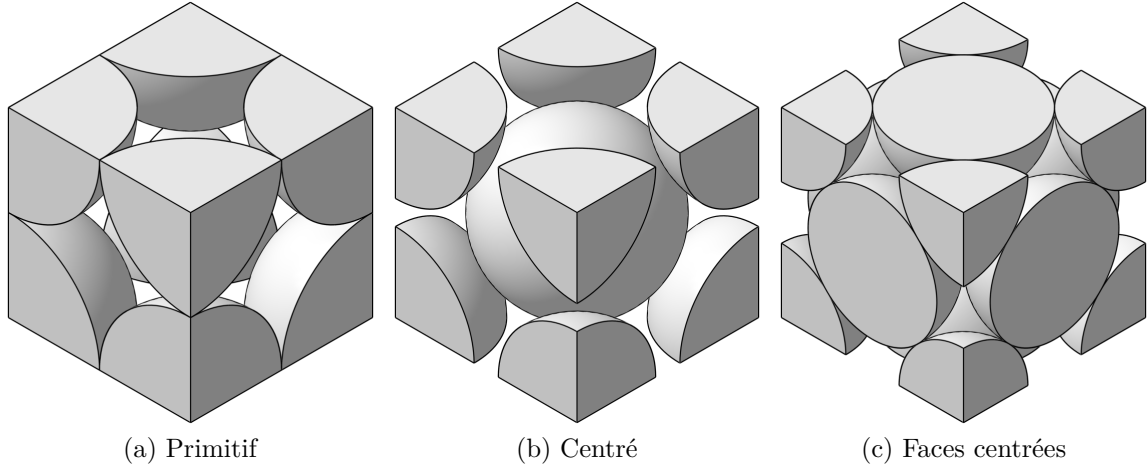


Figure 4.18 Géométrie des différents réseaux cubiques étudiés.

Dans le présent cas test, il n'y a pas de gradient de pression imposé et la force externe appliquée est constante (gravité). La loi de Darcy de l'équation (3.58) se simplifie donc ainsi :

$$q = \frac{\kappa}{\mu} \rho g = \frac{\kappa}{\nu} g \quad (4.68)$$

Le nombre de Darcy est ensuite introduit :

$$q = \text{Da} \frac{g d^2}{\nu} \quad (4.69)$$

La vitesse caractéristique U_0 de l'écoulement est ensuite définie en fonction du nombre de Darcy :

$$\text{Da} = \frac{q}{\frac{g d^2}{\nu}} = \frac{q}{U_0} \quad (4.70)$$

Où $U_0 := g d^2 / \nu$. Un écoulement incompressible avec force gravitationnelle est couramment caractérisé par le nombre de Reynolds et le nombre de Froude. Ceux-ci sont équivalents étant donné le U_0 choisi :

$$\text{Re} = \frac{U_0 L_0}{\nu} = \frac{g d^3}{\nu^2} \quad (4.71)$$

$$\text{Fr} = \frac{U_0^2}{g L_0} = \frac{g d^3}{\nu^2} \quad (4.72)$$

La quantité $\frac{g d^3}{\nu^2}$ obtenue correspond au rapport entre les forces gravitationnelles et les forces visqueuses. On décrit alors l'écoulement par le couple (Re, Da) plutôt que par le couple (Re, Fr). Les simulations sont effectuées dans un régime où le nombre de Reynolds est assez faible

pour que le nombre de Darcy soit indépendant de ce dernier, c'est-à-dire pour un écoulement rampant. On retrouve alors un unique nombre de Darcy pour une géométrie donnée.

Afin de comparer les résultats à ceux de la référence [40], le coefficient de traînée K tel que défini dans cette dernière est utilisé :

$$D = N K 3\pi\mu d U_{moy} \quad (4.73)$$

Où D est la traînée exercée sur les sphères et N correspond au nombre de sphères. Pour un écoulement rampant sur une seule sphère, on retrouve $K = 1$. Lorsque l'écoulement est en régime stationnaire, il est possible de lier le poids W du fluide et la force de traînée D à partir de la seconde loi de Newton :

$$D = W = \rho g \phi V \quad (4.74)$$

Où V est le volume de l'élément du milieu poreux considéré. Pour les réseaux de sphères, il est possible de lier la concentration de sphères N/V à la porosité :

$$1 - \phi = \frac{N \frac{1}{6} \pi d^3}{V} \quad (4.75)$$

On obtient alors une relation de proportionnalité entre la perméabilité adimensionnelle Da et le coefficient de traînée K .

$$K = \frac{1}{18} \frac{1}{(1 - \phi) Da} \quad (4.76)$$

La relation (4.76) démontre que pour le présent cas test, K et Da sont inversement proportionnels. Les résultats de référence tirée de la littérature étant exprimés en terme de K ; les résultats de simulation sont donc validés en comparant les coefficients de traînée K obtenus avec ceux de la littérature.

Le flux de Darcy numérique q_n est évalué numériquement lors des simulations à partir du flux massique moyen. Le flux massique moyen est calculé à partir de la moyenne sur toutes les cellules à deux pas de temps successifs. Ce choix est causé par le fait qu'une oscillation dans la solution perdure même en régime permanent. Cette oscillation est attribuée au fait que la paroi *bounce-back* est un schéma explicite en temps où le flux de particules réfléchi au pas de temps n ne dépend du pas du temps $n - 1$ mais dépend plutôt uniquement du flux de particules incident au pas de temps $n - 2$. Le flux de Darcy numérique est donc calculé ainsi :

$$q_n = \frac{\sum_i u_{i,n-1} \rho_{i,n-1} + \sum_i u_{i,n} \rho_{i,n}}{2 k \rho_0} \quad (4.77)$$

Où k correspond au nombre total de cellules dans le domaine, c'est-à-dire $k = N_L^3$, $\rho_0 = 1$ correspond à la masse volumique caractéristique et i correspond à l'indice des cellules du domaine. L'indice n correspond au pas de temps n et l'indice $n - 1$ correspond au pas de temps $n - 1$.

Le nombre de Darcy est donc évalué numériquement ainsi :

$$\text{Da}_n = \frac{q_n}{U_0} \quad (4.78)$$

Où U_0 est déterminé à partir de Re et des paramètres numériques τ , et N_L :

$$U_0 = \text{Re} \frac{\nu}{d} = \frac{\text{Re} \tau - 0.5}{3} \quad (4.79)$$

Le nombre de Reynolds a été choisi de manière à assurer un écoulement rampant où le nombre de Darcy est indépendant du nombre de Reynolds pour l'ensemble des simulations effectuées. Le critère d'arrêt est établi en observant les fluctuations en fonction du temps de la perméabilité évaluée numériquement. La méthode employée est la même que celle décrite à la section 4.3 pour la force de traînée.

Les conditions initiales sont fixées à vitesse nulle et masse volumique unitaire :

$$\mathbf{u}|_{t=0} = \mathbf{0} \quad (4.80)$$

$$\rho|_{t=0} = \rho_0 = 1 \quad (4.81)$$

4.5.1 Effet de l'opérateur de collision et du temps de relaxation sur la perméabilité

Le tableau 4.19a résume les paramètres de simulations employés. La figure 4.19 illustre le nombre de Darcy en fonction de la résolution spatiale pour le réseau de sphères primitif et pour différents opérateurs de collision et temps de relaxation. Sur cette figure, on remarque un bruit similaire sur chacune des courbes. Ce bruit correspond probablement aux erreurs de natures géométriques E_{disc} et E_{maill} décrites à la section 4.4. En effet, ces erreurs ne dépendent pratiquement pas de l'opérateur de collision et du temps de relaxation. Le bruit est causé par la discrétisation géométrique lors de la génération du maillage : elle induit des sauts dans la perméabilité calculée en fonction de la résolution.

Paramètre

Nombre de Reynolds (Re)	100
Modèle(s) de collision	BGK force externe, MRT force externe
Modèle(s) de condition aux frontières	<i>bounce-back</i>
Résolution spatiale (N_L)	$N_L = l = [35, 100]$
Résolution temporelle (N_t)	$N_t _{\tau=\{0.7,1,1.4\}} = \frac{N_L^2}{\text{Re}\nu} = \frac{3 N_L^2}{\text{Re}(\tau-0.5)}$

(a) Paramètres de simulation pour la section 4.5.1.

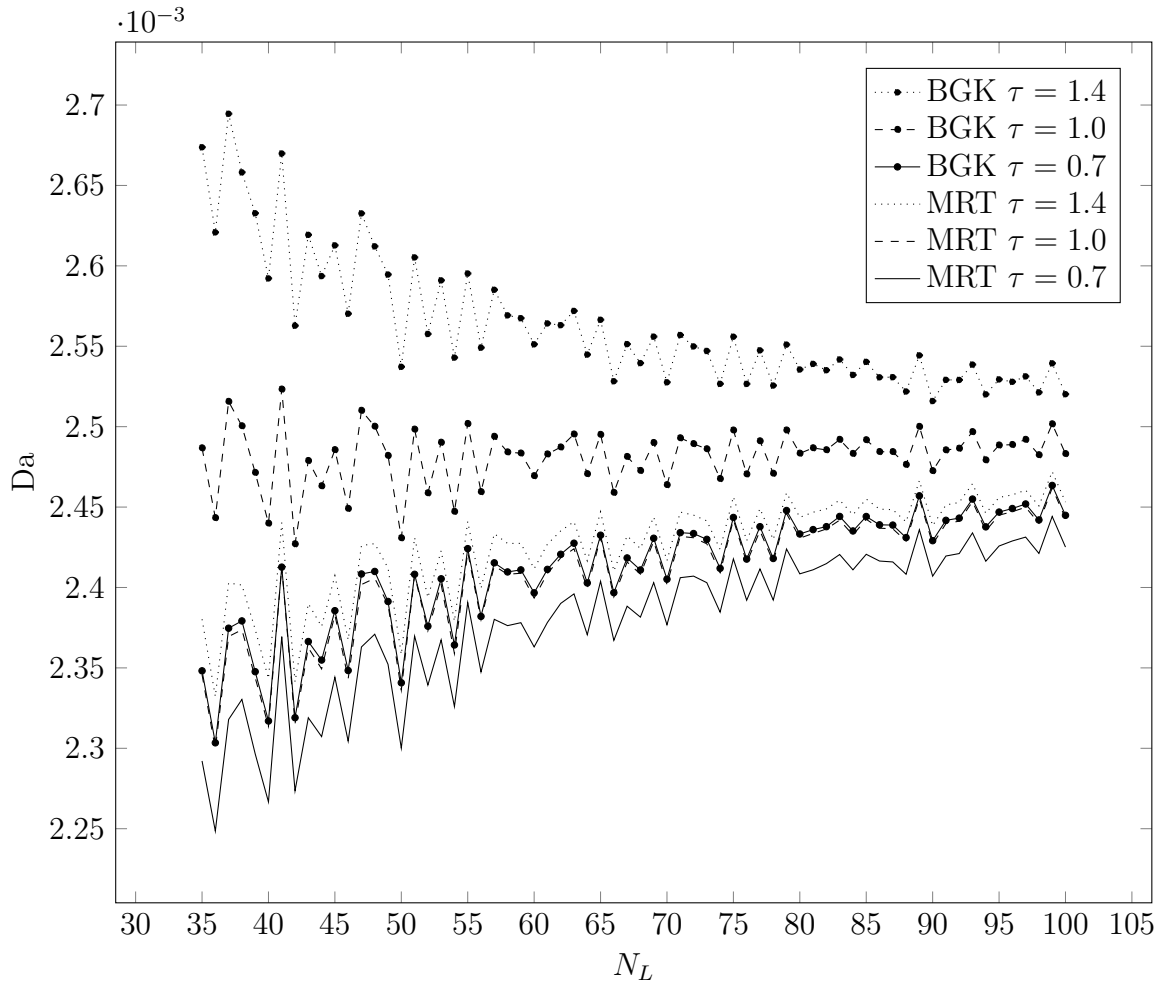
(b) Nombre de Darcy en fonction de la résolution pour différents τ et opérateurs de collision, pour le réseau primitif.

Figure 4.19 Effets de l'opérateur de collision et du temps de relaxation sur le nombre de Darcy lors du raffinement de maillage

On observe un décalage du comportement asymptotique entre chaque courbe probablement causé par E_{LB} , l'erreur de la méthode LB. On remarque que pour l'opérateur BGK, c'est le temps de relaxation $\tau = 1$ qui semble le moins sensible à la résolution spatiale N_L . Ceci est probablement causé par la position des parois *bounce-back* qui dépend du temps de relaxation [38]. Pour l'opérateur BGK, le temps de relaxation $\tau = 1$ semble le plus avantageux. Ceci pourrait être expliqué par le fait que ce choix de τ procure une position numérique des parois plus près de la position réelle. Autrement dit, la paroi numérique se trouverait directement à mi-chemin entre deux noeuds pour $\tau = 1$ [41]. Ceci est un phénomène étudié par plusieurs auteurs et depuis longtemps [42]. Ainsi, pour l'opérateur BGK, on conclut qu'un temps de relaxation plus faible (donc résolution temporelle plus fine) n'est pas nécessairement avantageux. Pour ce qui est du modèle MRT, on observe encore une fois qu'il est moins sensible au temps de relaxation. Comme au cas test de la section 4.4, il est possible d'utiliser cette insensibilité afin d'augmenter le temps de relaxation et atteindre le régime stationnaire plus rapidement.

4.5.2 Comparaison des perméabilités obtenues

Le tableau 4.20a illustre les paramètres employés afin de calculer le coefficient de traînée pour chacun des trois réseaux de sphères présentés. Le tableau 4.20b compare les résultats obtenus pour le calcul du coefficient K avec une référence de la littérature [40]. On constate que les résultats obtenus sont très près de ceux de cette référence. On rappelle que le coefficient K et la perméabilité Da sont inversement proportionnels, tel que démontré à l'équation (4.76).

Paramètre

Nombre de Reynolds (Re)	100
Modèle(s) de collision	MRT avec force externe
Modèle(s) de condition aux frontières	<i>bounce-back</i>
Résolution spatiale (N_L)	$N_L = l = 280$
Résolution temporelle (N_t)	$N_t _{\tau=0.7} = \frac{N_L^2}{\text{Re}\nu} = \frac{3 N_L^2}{\text{Re}(\tau-0.5)} = 11\,760$

(a) Paramètres de simulation pour la section 4.5.2.

Réseau	K prés.	K ref.
Primitif	42.3	42.1
Centré	163	162
Faces centrées	439	438

(b) Comparaison des K obtenues avec la référence [40].Figure 4.20 Comparaison des coefficients de traînée K obtenus pour différents réseaux cubiques

CHAPITRE 5 CONCLUSION

Synthèse des travaux

Le cadre théorique présenté au chapitre 2 permet de mettre en contexte certaines propriétés des fluides simulés. La faible compressibilité de la méthode ainsi que son équation d'état $p = \rho c_s^2$ proviennent de l'utilisation de la distribution de Maxwell-Boltzmann. Au chapitre 3, on constate que le fluide simulé hérite de certaines propriétés dues à la discrétisation. Celles-ci consistent entre autres en une vitesse du son constante, une relation entre τ et ν qui n'est pas directement proportionnel, mais uniquement proportionnel, et une approximation de la distribution de Maxwell-Boltzmann par un polynôme. Certaines de ces propriétés contraignent entre autres à simuler des écoulements incompressibles, malgré la compressibilité de la méthode toujours présente.

Contrairement aux équations de Navier-Stokes, il n'est pas simplificateur d'imposer l'incompressibilité et la stationnarité. En effet, il faudrait imposer des contraintes sur des moments de f , et ceci ne ferait que complexifier sa résolution. Il est plus simple d'utiliser la version compressible et instationnaire dans un régime où l'écoulement se comporte presque comme un écoulement incompressible et stationnaire. On constate également qu'on ne s'intéresse pas à résoudre f ou l'équation de Boltzmann. On s'intéresse plutôt à résoudre certains moments de f (moments physiques) ainsi que les équations de Navier-Stokes. Ceci laisse place à une certaine liberté lors de la discrétisation de l'équation de Boltzmann-BGK. Le modèle MRT, qui démontre certains avantages pour la simulation d'écoulements en milieux poreux, est un exemple de l'utilisation des moments non physiques de f à des fins numériques.

Dans le cadre des différents cas tests abordés, on a vérifié que la méthode LB permet de calculer les vitesses de l'écoulement, les forces causées par l'écoulement, ainsi que la perméabilité de milieux poreux. Durant les cas tests, une attention était apportée dans l'analyse des paramètres numériques afin de déterminer quel était le meilleur choix de ceux-ci. Pour ce qui est des conditions aux frontières, le compromis se situe principalement entre précision et stabilité. La paroi *bounce-back* est cependant beaucoup plus simple à implémenter, particulièrement pour les frontières non rectilignes, ce qui explique son utilisation pour les écoulements en milieux poreux. On conclut également que le choix du τ devrait dépendre du nombre de Reynolds. De plus, contrairement à la résolution spatiale, une trop forte augmentation de la résolution temporelle peut provoquer une instabilité, étant donné la contrainte de stabilité sur τ . Pour le cas du calcul de la perméabilité impliquant des écoulements rampants, le choix du τ devrait plutôt être fait dans l'optique d'accélérer le temps de calcul tout en considérant

son effet sur la position numérique des parois *bounce-back*.

Le quatrième cas test aborde un écoulement de Poiseuille 3D à travers un cylindre. Le but est de caractériser la principale source d'erreur lors de l'évaluation de la perméabilité d'un pore d'un milieu poreux. On observe que l'erreur est dominée par les effets géométriques, c'est-à-dire : la discrétisation de la géométrie, la représentation de surfaces courbes par des segments de surfaces perpendiculaires, et la position numérique des parois *bounce-back*. Ces effets géométriques sur l'erreur pourraient être approfondis pour des géométries poreuses plus réalistes. En effet, il serait intéressant de confirmer si, et à quel point, les effets géométriques dominent l'erreur pour le calcul de la perméabilité de géométries poreuses réelles. Peut-être qu'une étude de la méthode de génération du maillage permettrait d'estimer l'erreur de nature géométrique sur la perméabilité. Dans le cas où celle-ci est dominante, elle permettrait d'estimer l'erreur en fonction de la géométrie avant même d'effectuer des simulations.

Recommandations

Pour ce qui est de l'application en hydrologie concernant l'infiltration d'eau dans le sol, plusieurs améliorations pourraient être intégrées à la méthodologie présentée. Tout d'abord, un modèle diphasique (air et eau) en milieu poreux procurerait l'avantage de considérer le cas beaucoup plus commun pour lequel le sol n'est pas complètement saturé d'eau. L'utilisation de la méthode des éléments discrets permettrait de générer numériquement des milieux poreux granulaires en simulant le dépôt de granulats de formes sphériques ou de formes plus complexe. L'incorporation d'un modèle de transport de particules pourrait prendre en compte l'intrusion de particules fines qui provoquent une accumulation et un dépôt de matière solide à l'intérieur du milieu. On pourrait aussi s'intéresser à un régime dont le nombre de Reynolds est plus élevé, hors de la zone d'écoulement rampant. D'autre part, au niveau de l'implémentation informatique, le calcul sur GPU (*graphics processing unit*) pourrait pour sa part permettre d'utiliser des processeurs mieux adaptés pour le calcul en parallèle.

En somme, on constate deux directions intéressantes permettant d'approfondir les sujets abordés dans le présent mémoire. La première direction serait d'approfondir la caractérisation de l'erreur due aux effets géométriques lors d'écoulement en milieux poreux. Ceci serait particulièrement intéressant pour les géométries générées par tomодensitométrie. La tomодensitométrie permet de numériser un milieu poreux réel. Cependant, une erreur sur la position exacte des parois numérisées est commise. Il pourrait être pertinent de comparer la contribution de cette erreur par rapport à l'erreur commise par les parois de type *bounce-back*. Cette information permettrait de justifier ou non l'utilisation d'un modèle de conditions aux frontières plus précis. La seconde direction possible serait de se concentrer sur l'application en

hydrologie, telle que détaillée ci-haut. Les simulations au niveau microscopique permettraient d'améliorer les modèles hydrologiques. Il resterait cependant un travail à faire afin d'intégrer des résultats de simulations microscopiques à des modèles hydrologiques macroscopiques.

RÉFÉRENCES

- [1] Eleuterio.F Toro. *Riemann Solvers and Numerical Methods for Fluid Dynamics*. 2009.
- [2] OC Zienkiewicz, RL Taylor, and P Nithiarasu. *The finite element method for fluid dynamics*. 2005.
- [3] Sébastien Leclaire, Andrea Parmigiani, Orestis Malaspinas, Bastien Chopard, and Jonas Latt. Generalized three-dimensional lattice Boltzmann color-gradient method for immiscible two-phase pore-scale imbibition and drainage in porous media. *Physical Review E*, 95(3), 2017.
- [4] Michael C. Sukop and Daniel T. Thorne. *Lattice boltzmann modeling : An introduction for geoscientists and engineers*. 2006.
- [5] A. J C Ladd and R. Verberg. Lattice-Boltzmann simulations of particle-fluid suspensions. *Journal of Statistical Physics*, 104(5-6) :1191–1251, 2001.
- [6] David Vidal, Cathy Ridgway, Grégoire Pianet, Joachim Schoelkopf, Robert Roy, and François Bertrand. Effect of particle size distribution and packing compression on fluid permeability as predicted by lattice-Boltzmann simulations. *Computers and Chemical Engineering*, 33(1) :256–266, 2009.
- [7] Zhaoli Guo and T. S. Zhao. Lattice Boltzmann model for incompressible flows through porous media. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, 66(3) :1–9, 2002.
- [8] Sydney Chapman and T. G. Cowling. *The Mathematical Theory of Non-Uniform Gases*, volume 30. 1962.
- [9] Dominique D’Humières, Irina Ginzburg, Manfred Krafczyk, Pierre Lallemand, and Li-Shi Luo. Multiple-relaxation-time lattice Boltzmann models in three dimensions. *Philosophical Transactions of the Royal Society A : Mathematical, Physical and Engineering Sciences*, 360(1792) :437–451, 2002.
- [10] Erlend Magnus Viggen. *The lattice Boltzmann method : Fundamentals and acoustics*. PhD thesis, Norwegian University of Science and Technology, 2014.
- [11] Sauro Succi. *The Lattice Boltzmann Equation : For Fluid Dynamics and Beyond*. 2001.
- [12] P. L. Bhatnagar, E. P. Gross, and M. Krook. A model for collision processes in gases. I. Small amplitude processes in charged and neutral one-component systems. *Physical Review*, 94(3) :511–525, 1954.

- [13] Francois Golse. Chapter 3 The Boltzmann Equation and Its Hydrodynamic Limits. In *Handbook of Differential Equations : Evolutionary Equations*, volume 2, pages 159–301. 2006.
- [14] Bastien Chopard, Alexandre Dupuis, Alexandre Masselot, and Pascal Luthi. Cellular Automata and Lattice Boltzmann Techniques : An Approach to Model and Simulate Complex Systems. APPROACH TO MODEL AND SIMULATE COMPLEX SYSTEMS. *Advances in Complex Systems*, 05(02n03) :103–246, jun 2002.
- [15] M Gardner. Mathematical games : The fantastic combinations of John Conway’s new solitaire game "life". *Scientific American*, 223(October) :120–123., 1970.
- [16] U Frisch, B Hasslacher, and Y Pomeau. Lattice-gas automata for the Navier-Stokes equation. *Physical Review Letters*, 56(14) :1505–1508, apr 1986.
- [17] Guy R. McNamara and Gianluigi Zanetti. Use of the boltzmann equation to simulate lattice-gas automata. *Physical Review Letters*, 61(20) :2332–2335, 1988.
- [18] Xiaowen Shan, Xue-Feng Yuan, and Hudong Chen. Kinetic theory representation of hydrodynamics : a way beyond the Navier–Stokes equation. *Journal of Fluid Mechanics*, 550(-1) :413, 2006.
- [19] Xiaoyi He and Li-Shi Luo. Theory of the lattice Boltzmann method : From the Boltzmann equation to the lattice Boltzmann equation. *Physical Review E*, 56(6) :6811–6817, 1997.
- [20] Dieter a. Wolf-Gladrow. *Lattice-Gas Cellular Automata and Lattice Boltzmann Models - An Introduction*. 2005.
- [21] Zhaoli Guo, Chuguang Zheng, and Baochang Shi. Discrete lattice effects on the forcing term in the lattice Boltzmann method. *Physical Review E*, 65(4) :1–6, 2002.
- [22] Paul J. Dellar. Bulk and shear viscosities in lattice Boltzmann equations. *Physical Review E*, 64(3) :031203, 2001.
- [23] Frank White. *Fluid Mechanics*. 2010.
- [24] R J LeVeque. *Finite Difference Methods for Ordinary and Partial Differential Equations*. 2008.
- [25] FlowKit Ltd. Palabos.
- [26] Jonas Latt and Bastien Chopard. Lattice Boltzmann method with regularized pre-collision distribution functions. *Mathematics and Computers in Simulation*, 72(2-6) :165–168, sep 2006.

- [27] A. Kuzmin, Z. L. Guo, and A. A. Mohamad. Simultaneous incorporation of mass and force terms in the multi-relaxation-time framework for lattice Boltzmann schemes. *Philosophical Transactions of the Royal Society A : Mathematical, Physical and Engineering Sciences*, 369(1944) :2219–2227, 2011.
- [28] Jonas Latt, Bastien Chopard, Orestis Malaspinas, Michel Deville, and Andreas Michler. Straight velocity boundaries in the lattice Boltzmann method. *Physical Review E*, 77(5), 2008.
- [29] Qisu Zou and Xiaoyi He. On pressure and velocity boundary conditions for the lattice Boltzmann BGK model. *Physics of Fluids*, 9(6) :1591, 1997.
- [30] P. A. Skordos. Initial and boundary conditions for the lattice Boltzmann method. *Physical Review E*, 48(6) :4823–4842, 1993.
- [31] Takaji Inamuro, Masato Yoshino, and Fumimaru Ogino. A non-slip boundary condition for lattice Boltzmann simulations. *Physics of Fluids*, 7(12) :2928, 1995.
- [32] Jonas Lätt. *Hydrodynamic limit of lattice Boltzmann equations*. PhD thesis, University of Geneva, 2007.
- [33] Jonas Latt. Choice of units in lattice Boltzmann simulations, 2008.
- [34] Stephen Whitaker. Flow in porous media I : A theoretical derivation of Darcy’s law. *Transport in Porous Media*, 1(1) :3–25, 1986.
- [35] Ram Prakash Bharti, R. P. Chhabra, and V. Eswaran. Two-dimensional steady poiseuille flow of power-law fluids across a circular cylinder in a plane confined channel : Wall effects and drag coefficients. *Industrial and Engineering Chemistry Research*, 46(11) :3820–3840, 2007.
- [36] Alfonso Caiazzo and Michael Junk. Boundary forces in lattice Boltzmann : Analysis of Momentum Exchange algorithm. *Computers and Mathematics with Applications*, 55(7) :1415–1423, 2008.
- [37] Ismail B. Celik, Urmila Ghia, Patrick J. Roache, Christopher J. Freitas, Hugh Coleman, and Peter E. Raad. Procedure for Estimation and Reporting of Uncertainty Due to Discretization in CFD Applications. *Journal of Fluids Engineering*, 130(7) :78001, 2008.
- [38] Chongxun Pan, Li-Shi Luo, and Cass T. Miller. An evaluation of lattice Boltzmann schemes for porous medium flow simulation. *Computers & Fluids*, 35(8-9) :898–909, sep 2006.
- [39] Sébastien Leclaire. *Étude d’un modèle de Boltzmann sur réseau pour la simulation assistée par ordinateur des fluides à plusieurs phases immiscibles*. PhD thesis, 2013.

- [40] A. A. Zick and G. M. Homsy. Stokes flow through periodic arrays of spheres. *Journal of Fluid Mechanics*, 115 :13–26, 1982.
- [41] David Vidal, Robert Roy, and François Bertrand. A parallel workload balanced and memory efficient lattice-Boltzmann algorithm with single unit BGK relaxation time for laminar Newtonian flows. *Computers and Fluids*, 39(8) :1411–1423, 2010.
- [42] Xiaoyi He, Qisu Zou, Li-Shi Luo, and Micah Dembo. Analytic solutions of simple flows and analysis of nonslip boundary conditions for the lattice Boltzmann BGK model. *Journal of Statistical Physics*, 87(1-2) :115–136, 1997.
- [43] Eric W. Weisstein. Gaussian Integral.

ANNEXE A MOMENTS DE LA FONCTION DE DISTRIBUTION À L'ÉQUILIBRE f^{eq}

Dans l'annexe A, la masse volumique, la vitesse macroscopique et l'énergie cinétique de la fonction de distribution à l'équilibre f^{eq} sont calculés. Le cas en 3 dimensions est traité ; le cas en 2 dimensions est similaire. Afin d'abrégier les équations, les variables b et \mathbf{v} sont introduites :

$$b = \frac{1}{2 R T} \quad (\text{A.1})$$

$$\mathbf{v} = \mathbf{c} - \mathbf{u} \quad (\text{A.2})$$

$$d^3\mathbf{c} = dc_x dc_y dc_z = dv_x dv_y dv_z = d^3\mathbf{v} \quad (\text{A.3})$$

La fonction de distribution de l'équation (2.45) est réécrite ainsi :

$$f^{eq}(\mathbf{v}) = \rho \left(\frac{b}{\pi} \right)^{3/2} e^{-b|\mathbf{v}|^2} \quad (\text{A.4})$$

Pour l'évaluation des moments, les bornes des intégrales sont omises. Chaque intégrale est évaluée entre $-\infty$ et ∞ .

Masse volumique de f^{eq}

La masse volumique de la fonction de distribution à l'équilibre est calculée à partir des équations (2.45) et (2.19) :

$$\rho_{eq} = \int f^{eq} d^3\mathbf{c} = \int \rho \left(\frac{b}{\pi} \right)^{3/2} e^{-b|\mathbf{v}|^2} d^3\mathbf{v} \quad (\text{A.5})$$

On développe $|\mathbf{v}|^2$ et $d^3\mathbf{c}$:

$$\rho_{eq} = \iiint \rho \left(\frac{b}{\pi} \right)^{3/2} e^{-b(v_x^2 + v_y^2 + v_z^2)} dv_x dv_y dv_z \quad (\text{A.6})$$

Par linéarité :

$$\rho_{eq} = \rho \left(\frac{b}{\pi} \right)^{3/2} \int e^{-bv_x^2} dc_x \int e^{-bv_y^2} dc_y \int e^{-bv_z^2} dv_z \quad (\text{A.7})$$

L'intégrale de la forme suivante est traité séparément :

$$\int e^{-bv_i^2} dv_i \quad (\text{A.8})$$

On effectue un changement de variable :

$$w_i = \sqrt{b}v_i \quad (\text{A.9})$$

$$dv_i = \frac{dw_i}{\sqrt{b}} \quad (\text{A.10})$$

$$\frac{1}{\sqrt{b}} \int e^{-w_i^2} dw_i \quad (\text{A.11})$$

On introduit le résultat de l'intégrale de Gauss [43] :

$$\int_{-\infty}^{\infty} e^{-x^2} dx = \sqrt{\pi} \quad (\text{A.12})$$

On obtient alors :

$$\int e^{-bv_i^2} dv_i = \sqrt{\frac{\pi}{b}} \quad (\text{A.13})$$

Par substitution de l'équation (A.13) dans l'équation (A.7), on obtient :

$$\rho_{eq} = \rho \left(\frac{b}{\pi} \right)^{3/2} \left(\sqrt{\frac{\pi}{b}} \right)^3 = \rho \quad (\text{A.14})$$

Vitesse macroscopique de f^{eq}

La vitesse macroscopique de la fonction de distribution à l'équilibre est calculée à partir des équations (2.45) et (2.20) :

$$\rho_{eq} \mathbf{u}_{eq} = \int f^{eq} \mathbf{c} d^3 \mathbf{c} = \int \rho \left(\frac{b}{\pi} \right)^{3/2} e^{-b|\mathbf{v}|^2} (\mathbf{v} + \mathbf{u}) d^3 \mathbf{v} \quad (\text{A.15})$$

On développe $|\mathbf{v}|^2$ et $d^3 \mathbf{c}$:

$$\rho_{eq} \mathbf{u}_{eq} = \rho \left(\frac{b}{\pi} \right)^{3/2} \iiint e^{-bv_x^2} e^{-bv_y^2} e^{-bv_z^2} (\mathbf{v} + \mathbf{u}) dv_x dv_y dv_z \quad (\text{A.16})$$

Seule la composante u_x de \mathbf{u} est traitée, les deux autres composantes se traitent de manière similaire. Par linéarité, on obtient :

$$\rho_{eq} u_{x eq} = \rho \left(\frac{b}{\pi} \right)^{3/2} \int e^{-bv_x^2} (v_x + u_x) dv_x \int e^{-bv_y^2} dv_y \int e^{-bv_z^2} dv_z \quad (\text{A.17})$$

L'intégrale de la forme suivante est traité séparément :

$$\int e^{-bv_i^2} (v_i + u_i) dv_i \quad (\text{A.18})$$

Par linéarité :

$$\int e^{-bv_i^2} (v_i + u_i) dv_i = \int e^{-bv_i^2} v_i dv_i + u_i \int e^{-bv_i^2} dv_i \quad (\text{A.19})$$

Le membre de gauche de l'équation (A.19) est nulle puisque c'est l'intégrale d'une fonction antisymétrique. Ceci est prouvé en tentant de résoudre l'intégrale suivante :

$$\int e^{-bv_i^2} v_i dv_i \quad (\text{A.20})$$

On effectue un changement de variable :

$$w_i = -bv_i^2 \quad (\text{A.21})$$

$$dw_i = -2bv_i dv_i \quad (\text{A.22})$$

Ce qui permet de résoudre l'intégrale (A.20) :

$$\int e^{-bv_i^2} v_i dv_i = -\frac{1}{2b} \int e^{w_i} dw_i = -\frac{1}{2b} [e^{w_i}]_{v_i=-\infty}^{v_i=\infty} = -\frac{1}{2b} [e^{-v_i^2}]_{v_i=-\infty}^{v_i=\infty} \quad (\text{A.23})$$

On retrouve enfin que le premier terme de l'équation (A.19) est nulle :

$$\int e^{-bv_i^2} v_i \, dv_i = -\frac{1}{2b} \left[e^{-v_i^2} \right]_{v_i=-\infty}^{v_i=\infty} = -\frac{1}{2b} \left[e^{-v_i^2} \right]_{v_i=-\infty}^{v_i=0} - \frac{1}{2b} \left[e^{-v_i^2} \right]_{v_i=0}^{v_i=\infty} = 0 \quad (\text{A.24})$$

$$\int e^{-bv_i^2} (v_i + u_i) \, dv_i = u_i \sqrt{\frac{\pi}{b}} \quad (\text{A.25})$$

En substituant les équations (A.13) et (A.25) dans l'équation (A.17), on obtient la composante x de la vitesse macroscopique :

$$\rho_{eq} u_{xeq} = \rho \left(\frac{b}{\pi} \right)^{3/2} u_x \left(\frac{\sqrt{\pi}}{\sqrt{b}} \right)^3 = \rho u_x \quad (\text{A.26})$$

On obtient de manière similaire que $\rho_{eq} u_{yeq} = \rho u_y$ et $\rho_{eq} u_{zeq} = \rho u_z$. On conclut alors que $\rho_{eq} \mathbf{u}_{eq} = \rho \mathbf{u}$.

Énergie cinétique massique de f^{eq}

L'énergie cinétique massique de la fonction de distribution à l'équilibre est calculée à partir des équations (2.45) et (2.22) :

$$\rho_{eq} \epsilon_{eq} = \frac{1}{2} \int f^{eq} |\mathbf{c} - \mathbf{u}|^2 \, d^3 \mathbf{c} = \frac{1}{2} \int \rho \left(\frac{b}{\pi} \right)^{3/2} e^{-b|\mathbf{v}|^2} |\mathbf{v}|^2 \, d^3 \mathbf{v} \quad (\text{A.27})$$

On développe $|\mathbf{v}|^2$ et $d^3 \mathbf{c}$:

$$\rho_{eq} \epsilon_{eq} = \frac{1}{2} \rho \left(\frac{b}{\pi} \right)^{3/2} \int e^{-b(v_x^2 + v_y^2 + v_z^2)} (v_x^2 + v_y^2 + v_z^2) \, d^3 \mathbf{v} \quad (\text{A.28})$$

Par linéarité :

$$\begin{aligned} \rho_{eq} \epsilon_{eq} &= \int e^{-bv_x^2} v_x^2 \, dv_x \int e^{-bv_y^2} \, dv_y \int e^{-bv_z^2} \, dv_z \\ &\quad + \int e^{-bv_x^2} \, dv_x \int e^{-bv_y^2} v_y^2 \, dv_y \int e^{-bv_z^2} \, dv_z \\ &\quad + \int e^{-bv_x^2} \, dv_x \int e^{-bv_y^2} \, dv_y \int e^{-bv_z^2} v_z^2 \, dv_z \end{aligned} \quad (\text{A.29})$$

L'intégrale de la forme suivante est traitée séparément :

$$\int e^{-bv_i^2} v_i^2 dv_i \quad (\text{A.30})$$

On effectue une intégration par parties :

$$g = v_i \quad h' = v_i e^{-bv_i^2} \quad (\text{A.31})$$

$$g' = 1 \quad h = -\frac{1}{2b} e^{-bv_i^2} \quad (\text{A.32})$$

$$\int e^{-bv_i^2} v_i^2 dv_i = \left[-\frac{v_i}{2b} e^{-bv_i^2} \right]_{v_i=-\infty}^{v_i=\infty} + \frac{1}{2b} \int e^{-bv_i^2} dv_i \quad (\text{A.33})$$

Le terme de droite du membre de droite est déterminée par l'équation (A.13). Le terme de gauche du membre de droite est évalué à partir de la règle de L'Hôpital :

$$\lim_{v_i \rightarrow \infty} -\frac{v_i}{2b} e^{-bv_i^2} = \lim_{v_i \rightarrow \infty} -\frac{v_i}{2be^{bv_i^2}} = \lim_{v_i \rightarrow \infty} -\frac{1}{4b^2 v_i e^{bv_i^2}} = 0 \quad (\text{A.34})$$

$$\lim_{v_i \rightarrow -\infty} -\frac{v_i}{2b} e^{-bv_i^2} = \lim_{v_i \rightarrow -\infty} -\frac{v_i}{2be^{bv_i^2}} = \lim_{v_i \rightarrow -\infty} -\frac{1}{4b^2 v_i e^{bv_i^2}} = 0 \quad (\text{A.35})$$

On retrouve donc :

$$\int e^{-bv_i^2} v_i^2 dv_i = \frac{1}{2b} \sqrt{\frac{\pi}{b}} \quad (\text{A.36})$$

En substituant les équations (A.13), et (A.36) dans l'équation (A.29), on obtient l'énergie cinétique massique :

$$\rho_{eq} \epsilon_{eq} = \frac{1}{2} \rho \left(\frac{b}{\pi} \right)^{3/2} 3 \frac{1}{2b} \left(\sqrt{\frac{\pi}{b}} \right)^3 = \rho \frac{3}{2} RT \quad (\text{A.37})$$

ANNEXE B DÉTAIL DU CODE D'UNE SIMULATION

Le code suivant est celui utilisé afin de générer les résultats de simulations pour le cas test de la section 4.5 pour l'écoulement à travers des réseaux de sphères. Pour plus d'informations concernant l'utilisation de la librairie Palabos, se référer au site internet officiel : <http://www.palabos.org/>

```

1  #include "palabos3D.h"
2  #include "palabos3D.hh"
3  #include <vector>
4  #include <cmath>
5  #include <iostream>
6  #include <fstream>
7  #include <ctime>
8  #include <string>
9  #include <sys/stat.h>
10 #include <limits>
11
12 using namespace plb;
13 using namespace std;
14
15 typedef double T;
16
17 // Macro permettant de choisir le modele de collision BGK ou MRT
18 // #define DYNAMICS GuoExternalForceBGKdynamics
19 #define DYNAMICS GuoExternalForceMRTdynamics
20
21 // Macro permettant de choisir le modele schema DnQm
22 // #define DESCRIPTOR descriptors::ForcedD3Q19Descriptor
23 #define DESCRIPTOR descriptors::ForcedMRTD3Q19Descriptor
24
25 // La classe ForcedFlowParam contient les parametres de l'écoulement
26 template <typename T>
27 class ForcedFlowParam {
28     public:
29         ForcedFlowParam(T Ga_, T tau_, T L0_, plint Nx_)

```

```

30         : Ga(Ga_), tau(tau_), L0(L0_), Nx(Nx_) {}
31 // Longueur caracteristique L0 (diametre des spheres)
32 T getL0() const { return L0; }
33 // Nombre de Galilee ( $g \cdot L^3 / \nu^2$ )
34 T getGa() const { return Ga; }
35 // Resolution spatiale
36 plint getNx() const { return Nx; }
37 // Temps de relaxation
38 T getTau() const { return tau; }
39 // Pas d'espace adimensionnel
40 T getDeltaX() const { return 1. / (T)getL0(); }
41 // Viscosite
42 T getLatticeNu() const { return (getTau() - 0.5) / 3.; }
43 // Frequence de relaxation
44 T getOmega() const { return (T)1. / getTau(); }
45 // Force appliquee
46 T getForce() const {
47     return getGa() * pow(getLatticeNu(), 2.) / pow((T)getL0(), 3.);
48 }
49 // Vitesse caracteristique U0
50 T getLatticeU() const {
51     return pow((T)getL0(), 2.) * getForce() / getLatticeNu();
52 }
53 // Pas de temps adimensionnel
54 T getDeltaT() const { return getDeltaX() / getLatticeU(); }
55
56 private:
57     T Ga, tau, L0;
58     plint Nx;
59 };
60
61 // La classe GeoParam contient les parametres relatifs a la
62 // geometrie
63 template <typename T>
64 class GeoParam {
65 public:
66     GeoParam(T cx_, T cy_, T cz_, T L0_, plint Nx_, plint reseau_)
67         : cx(cx_), cy(cy_), cz(cz_), L0(L0_), Nx(Nx_), reseau(reseau_)

```

```

    {}
67 // Type de reseau (1 = cubique , 2 = centre , 3 = faces centrees)
68 T getReseau() const { return reseau; }
69 // Coordonnees du centre du reseau par rapport au centre d'une
    cellule
70 T getCx() const { return cx; }
71 T getCy() const { return cy; }
72 T getCz() const { return cz; }
73 // Resolution spatiale
74 plint getNx() const { return Nx; }
75 // Longueur caracteristique L0 (diametre des spheres)
76 T getL0() const { return L0; }
77
78 private:
79     T cx, cy, cz, L0;
80     plint Nx, reseau;
81 };
82
83 // La classe permet d'initialiser la force externe et les conditions
    initiales
84 template <typename T, template <typename U> class Descriptor>
85 class ExternalForceInitializer
86     : public OneCellIndexedFunctional3D<T, Descriptor> {
87 public:
88     ExternalForceInitializer(T force_) : force(force_) {}
89     ExternalForceInitializer<T, Descriptor>* clone() const {
90         return new ExternalForceInitializer<T, Descriptor>(*this);
91     }
92     virtual void execute(plint iX, plint iY, plint iZ,
93                         Cell<T, Descriptor>& cell) const {
94         T force_x = force;
95         T force_y = 0;
96         T force_z = 0;
97         Array<T, 3> forcefield(force_x, force_y, force_z);
98         Array<T, 3> zeroVelocity(0., 0., 0.);
99         T rho = (T)1.;
100
101         // Conditions initiale rho = 1 et u = (0,0,0)

```



```

102     iniCellAtEquilibrium(cell, rho, zeroVelocity);
103     // Imposition de la force externe
104     cell.setExternalField(Descriptor<T>::ExternalField::
        forceBeginsAt,
105                             Descriptor<T>::ExternalField::sizeOfForce,
106                             &forcefield[0]);
107 }
108
109 private:
110     T force;
111 };
112
113 // La classe CubicPackingDomain3D permet de generer la geometrie du
    reseau
114 // cubique
115 template <typename T>
116 class CubicPackingDomain3D : public DomainFunctional3D {
117 public:
118     CubicPackingDomain3D(GeoParam<T> const& geoparameters_, T delta_)
119         : geoparameters(geoparameters_), delta(delta_) {
120         cx = geoparameters.getCx();
121         cy = geoparameters.getCy();
122         cz = geoparameters.getCz();
123         sqrdR = pow(geoparameters.getL0() / 2 - delta, 2.);
124         Nx = geoparameters_.getNx();
125     }
126     virtual bool operator()(plint iX, plint iY, plint iZ) const {
127         return (pow(fmod(iX + Nx / 2. - cx, (T)Nx) - Nx / 2., 2) +
128                 pow(fmod(iY + Nx / 2. - cy, (T)Nx) - Nx / 2., 2) +
129                 pow(fmod(iZ + Nx / 2. - cz, (T)Nx) - Nx / 2., 2)) <=
                    sqrdR;
130     }
131     virtual CubicPackingDomain3D<T>* clone() const {
132         return new CubicPackingDomain3D<T>(*this);
133     }
134
135 private:
136     GeoParam<T> const& geoparameters;

```

```

137   T cx, cy, cz, sqrdR, delta;
138   plint Nx;
139 };
140
141 // La classe CenteredPackingDomain3D permet de generer la geometrie
    du reseau
142 // cubique centre
143 template <typename T>
144 class CenteredPackingDomain3D : public DomainFunctional3D {
145 public:
146   CenteredPackingDomain3D(GeoParam<T> const& geoparameters_, T
        delta_)
147       : geoparameters(geoparameters_), delta(delta_) {
148     cx = geoparameters.getCx();
149     cy = geoparameters.getCy();
150     cz = geoparameters.getCz();
151     sqrdR = pow(geoparameters.getL0() / 2 - delta, 2.);
152     Nx = geoparameters_.getNx();
153   }
154   virtual bool operator()(plint iX, plint iY, plint iZ) const {
155     return ((pow(fmod(iX + Nx - cx, (T)Nx) - Nx / 2., 2) +
156             pow(fmod(iY + Nx - cy, (T)Nx) - Nx / 2., 2) +
157             pow(fmod(iZ + Nx - cz, (T)Nx) - Nx / 2., 2)) <= sqrdR)
158           ||
159           ((pow(fmod(iX + Nx / 2. - cx, (T)Nx) - Nx / 2., 2) +
160             pow(fmod(iY + Nx / 2. - cy, (T)Nx) - Nx / 2., 2) +
161             pow(fmod(iZ + Nx / 2. - cz, (T)Nx) - Nx / 2., 2)) <=
                sqrdR);
162   }
163   virtual CenteredPackingDomain3D<T>* clone() const {
164     return new CenteredPackingDomain3D<T>(*this);
165   }
166 private:
167   GeoParam<T> const& geoparameters;
168   T cx, cy, cz, sqrdR, delta;
169   plint Nx;
170 };

```

```

171
172 // La classe FaceCenteredPackingDomain3D permet de generer la
    geometrie du
173 // reseau cubique a faces centrees
174 template <typename T>
175 class FaceCenteredPackingDomain3D : public DomainFunctional3D {
176 public:
177     FaceCenteredPackingDomain3D(GeoParam<T> const& geoparameters_, T
        delta_)
178         : geoparameters(geoparameters_), delta(delta_) {
179         cx = geoparameters.getCx();
180         cy = geoparameters.getCy();
181         cz = geoparameters.getCz();
182         sqrdR = pow(geoparameters.getL0() / 2 - delta, 2.);
183         Nx = geoparameters_.getNx();
184     }
185     virtual bool operator()(plint iX, plint iY, plint iZ) const {
186         return ((pow(fmod(iX + Nx / 2. - cx, (T)Nx) - Nx / 2., 2) +
187             pow(fmod(iY + Nx / 2. - cy, (T)Nx) - Nx / 2., 2) +
188             pow(fmod(iZ + Nx / 2. - cz, (T)Nx) - Nx / 2., 2)) <=
                sqrdR) ||
189             ((pow(fmod(iX + Nx / 2. - cx, (T)Nx) - Nx / 2., 2) +
190             pow(fmod(iY + Nx - cy, (T)Nx) - Nx / 2., 2) +
191             pow(fmod(iZ + Nx - cz, (T)Nx) - Nx / 2., 2)) <= sqrdR)
                ||
192             ((pow(fmod(iX + Nx - cx, (T)Nx) - Nx / 2., 2) +
193             pow(fmod(iY + Nx / 2. - cy, (T)Nx) - Nx / 2., 2) +
194             pow(fmod(iZ + Nx - cz, (T)Nx) - Nx / 2., 2)) <= sqrdR)
                ||
195             ((pow(fmod(iX + Nx - cx, (T)Nx) - Nx / 2., 2) +
196             pow(fmod(iY + Nx - cy, (T)Nx) - Nx / 2., 2) +
197             pow(fmod(iZ + Nx / 2. - cz, (T)Nx) - Nx / 2., 2)) <=
                sqrdR);
198     }
199     virtual FaceCenteredPackingDomain3D<T>* clone() const {
200         return new FaceCenteredPackingDomain3D<T>(*this);
201     }
202

```

```

203 private:
204     GeoParam<T> const& geoparameters;
205     T cx, cy, cz, sqrdR, delta;
206     plint Nx;
207 };
208
209 // Classe assignant les noeuds solides du reseau cubique a partir de
    sa
210 // geometrie
211 template <typename T>
212 class SolidNodes1 {
213 public:
214     SolidNodes1(GeoParam<T> const& geoparameters_)
215         : geoparameters(geoparameters_) {}
216     bool operator()(plint iX, plint iY, plint iZ) const {
217         return !CubicPackingDomain3D<T>(geoparameters, 0.)(iX, iY, iZ);
218     }
219
220 private:
221     GeoParam<T> const& geoparameters;
222 };
223
224 // Classe assignant les noeuds solides du reseau centre a partir de
    sa geometrie
225 template <typename T>
226 class SolidNodes2 {
227 public:
228     SolidNodes2(GeoParam<T> const& geoparameters_)
229         : geoparameters(geoparameters_) {}
230     bool operator()(plint iX, plint iY, plint iZ) const {
231         return !CenteredPackingDomain3D<T>(geoparameters, 0.)(iX, iY, iZ
        );
232     }
233
234 private:
235     GeoParam<T> const& geoparameters;
236 };
237

```

```

238 // Classe assignant les noeuds solides du reseau a faces centrees a
      partir de sa
239 // geometrie
240 template <typename T>
241 class SolidNodes3 {
242 public:
243     SolidNodes3(GeoParam<T> const& geoparameters_)
244         : geoparameters(geoparameters_) {}
245     bool operator()(plint iX, plint iY, plint iZ) const {
246         return !FaceCenteredPackingDomain3D<T>(geoparameters, 0.)(iX, iY
            , iZ);
247     }
248
249 private:
250     GeoParam<T> const& geoparameters;
251 };
252
253 // Creation d'un fichier de sortie pour les champs de densite (
      pression) et
254 // vitesse
255 void writeVTK(MultiBlockLattice3D<T, DESCRIPTOR>& lattice,
256               ForcedFlowParam<T> const& parameters, plint iter) {
257     T dx = parameters.getDeltaX();
258     T dt = parameters.getDeltaT();
259     VtkImageOutput3D<T> vtkOut(createFileName("vtk", iter, 6), dx);
260     vtkOut.writeData<float>(*computeDensity(lattice), "density", 1);
261     vtkOut.writeData<3, float>(*computeVelocity(lattice), "velocity",
        dx / dt);
262 }
263
264 // Calcul de la porosite pour le reseau cubique
265 T computePorosity1(GeoParam<T> const& geoparameters) {
266     MultiScalarField3D<T> flagMatrix(geoparameters.getNx(),
        geoparameters.getNx(),
267                                       geoparameters.getNx());
268     setToFunction(flagMatrix, flagMatrix.getBoundingBox(),
269                  SolidNodes1<T>(geoparameters));
270     return computeAverage(flagMatrix);

```

```

271 }
272
273 // Calcul de la porosite pour le reseau centre
274 T computePorosity2(GeoParam<T> const& geoparameters) {
275     MultiScalarField3D<T> flagMatrix(geoparameters.getNx(),
276                                     geoparameters.getNx(),
277                                     geoparameters.getNx());
278     setToFunction(flagMatrix, flagMatrix.getBoundingBox(),
279                 SolidNodes2<T>(geoparameters));
279     return computeAverage(flagMatrix);
280 }
281
282 // Calcul de la porostie pour le reseau a faces centrees
283 T computePorosity3(GeoParam<T> const& geoparameters) {
284     MultiScalarField3D<T> flagMatrix(geoparameters.getNx(),
285                                     geoparameters.getNx(),
286                                     geoparameters.getNx());
287     setToFunction(flagMatrix, flagMatrix.getBoundingBox(),
288                 SolidNodes3<T>(geoparameters));
288     return computeAverage(flagMatrix);
289 }
290
291 // Calcul du flux massique moyen dans le sens de la force imposee
292 T computeAvgMassFlux(MultiBlockLattice3D<T, DESCRIPTOR>& lattice) {
293     return computeAverage(
294         *multiply(*computeVelocityComponent(lattice, lattice.
295             getBoundingBox(), 0),
296             *computeDensity(lattice, lattice.getBoundingBox()))
297         );
296 }
297 // Calcul de la vitesse moyenne dans le sens de la force imposee
298 T computeAvgVelComp(MultiBlockLattice3D<T, DESCRIPTOR>& lattice) {
299     return computeAverage(
300         *computeVelocityComponent(lattice, lattice.getBoundingBox(),
301             0));
301 }
302
303 // Debut du programme principal

```

```

304 int main(int argc, char* argv[]) {
305     plbInit(&argc, &argv);
306     // Creation d'un dossier pour stocker les resultats de simulation
307     char dir[28];
308     time_t now = time(NULL);
309     strftime(dir, 28, "./tmp/%Y-%m-%d_%H.%M%S/", localtime(&now));
310     mkdir(dir, S_IRWXU | S_IRWXG | S_IROTH | S_IXOTH);
311     global::directories().setOutputDir(dir);
312
313     plint vtkSave;
314     T r, maxT, logT, epsilon;
315     // Ouverture du fichier de parametres supplementaires
316     try {
317         XMLreader xmlFile("inputParameters.xml");
318         xmlFile["Simulation"]["maxT"].read(maxT);
319         xmlFile["Simulation"]["epsilon"].read(epsilon);
320         xmlFile["Simulation"]["r"].read(r);
321         xmlFile["Output"]["logT"].read(logT);
322         xmlFile["Output"]["vtkSave"].read(vtkSave);
323     } catch (PlbIOException& exception) {
324         pcout << exception.what() << endl;
325         return -1;
326     }
327
328     // Lecture des parametres d'entree
329     if (argc != 8) {
330         pcout << "Erreur, il manque des parametres d'entree \n";
331         pcout << "La structure est :\n";
332         pcout << "1. Ga = g * L^3 * nu^-2\n";
333         pcout << "2. Nx \n";
334         pcout << "3. tau \n";
335         pcout << "4. type reseau \n";
336         pcout << "5. cx \n";
337         pcout << "6. cy \n";
338         pcout << "7. cz \n";
339         pcout << "Example: " << argv[0] << " 0.005 30 2 1 0 0 0\n";
340         exit(EXIT_FAILURE);
341     }

```

```

342
343 // Conversion des parametres d'entree
344 const plint reseau = atoi(argv[4]);
345 const T cx = atof(argv[5]);
346 const T cy = atof(argv[6]);
347 const T cz = atof(argv[7]);
348 T ratio, poro;
349 // Calcul du ratio entre la largeur du domaine de calcul et le
    diametre d'une
350 // sphere L0
351 if (reseau == 1) {
352     ratio = r * sqrt(4) / 4.;
353 } else if (reseau == 2) {
354     ratio = r * sqrt(3) / 4.;
355 } else if (reseau == 3) {
356     ratio = r * sqrt(2) / 4.;
357 } else {
358     pcout << "Mauvais input reseau de spheres\n";
359     exit(EXIT_FAILURE);
360 }
361
362 // Conversion des parametres d'entree
363 ForcedFlowParam<T> parameters(atof(argv[1]),           // Ga
364                               atof(argv[3]),           // tau
365                               atof(argv[2]) * 2 * ratio, // L0
366                               atoi(argv[2])            // Nx
367                               );
368
369 // Assignation des parametres geometriques
370 GeoParam<T> geoparameters(cx, cy, cz, r, parameters.getL0(),
371                             parameters.getNx(), reseau);
372
373 // Initiatlisation de la matrice stockant les populations de
    particules de
374 // chaque noeud.
375 // Les noeuds sont attribues a differents processeurs pour la
    parallelisation
376 MultiBlockLattice3D<T, DESCRIPTOR> lattice(

```



```

377     parameters.getNx(), parameters.getNx(), parameters.getNx(),
378     new DYNAMICS<T, DESCRIPTOR>(parameters.getOmega()));
379
380 // Initialisation de la force calculee sur les spheres
381 Array<plint, 3> forceIds;
382 forceIds[0] = lattice.internalStatSubscription().subscribeSum();
383 forceIds[1] = lattice.internalStatSubscription().subscribeSum();
384 forceIds[2] = lattice.internalStatSubscription().subscribeSum();
385
386 // Attribution de periodicite a toutes les frontieres
387 lattice.periodicity().toggleAll(true);
388
389 // Creation de la geometrie
390 applyIndexed(
391     lattice, lattice.getBoundingBox(),
392     new ExternalForceInitializer<T, DESCRIPTOR>(parameters.
393         getForce()));
394 if (reseau == 1) {
395     defineDynamics(lattice, lattice.getBoundingBox(),
396         new CubicPackingDomain3D<T>(geoparameters, 0.),
397         new MomentumExchangeBounceBack<T, DESCRIPTOR>(
398             forceIds));
399     defineDynamics(lattice, lattice.getBoundingBox(),
400         new CubicPackingDomain3D<T>(geoparameters, sqrt
401             (2) + 1E-6),
402         new NoDynamics<T, DESCRIPTOR>());
403     initializeMomentumExchange(lattice, lattice.getBoundingBox(),
404         new CubicPackingDomain3D<T>(
405             geoparameters, 0.));
406     poro = computePorosity1(geoparameters);
407 } else if (reseau == 2) {
408     defineDynamics(lattice, lattice.getBoundingBox(),
409         new CenteredPackingDomain3D<T>(geoparameters, 0.)
410         ,
411         new MomentumExchangeBounceBack<T, DESCRIPTOR>(
412             forceIds));
413     defineDynamics(
414         lattice, lattice.getBoundingBox(),

```

```

409         new CenteredPackingDomain3D<T>(geoparameters, sqrt(2) + 1E
            -6),
410         new NoDynamics<T, DESCRIPTOR>());
411     initializeMomentumExchange(
412         lattice, lattice.getBoundingBox(),
413         new CenteredPackingDomain3D<T>(geoparameters, 0.));
414     poro = computePorosity2(geoparameters);
415 } else if (reseau == 3) {
416     defineDynamics(lattice, lattice.getBoundingBox(),
417         new FaceCenteredPackingDomain3D<T>(geoparameters,
            0.),
418         new MomentumExchangeBounceBack<T, DESCRIPTOR>(
            forceIds));
419     defineDynamics(
420         lattice, lattice.getBoundingBox(),
421         new FaceCenteredPackingDomain3D<T>(geoparameters, sqrt(2) +
            1E-6),
422         new NoDynamics<T, DESCRIPTOR>());
423     initializeMomentumExchange(
424         lattice, lattice.getBoundingBox(),
425         new FaceCenteredPackingDomain3D<T>(geoparameters, 0.));
426     poro = computePorosity3(geoparameters);
427 } else {
428     pcout << "Mauvais input reseau de spheres\n";
429     exit(EXIT_FAILURE);
430 }
431
432 // Initialisation de la geometrie
433 lattice.initialize();
434
435 const plint maxiT =
436     (plint)ceil(maxT * 250000 / pow((T)parameters.getNx(), 3.)) *
        4;
437 const plint logiT =
438     (plint)ceil(logT * 250000 / pow((T)parameters.getNx(), 3.)) *
        4;
439 const plint vtkSaveiT =
440     (plint)ceil(vtkSave * 250000 / pow((T)parameters.getNx(), 3.))

```

```

        * 4;
441  const plint logVT = (plint) ceil((T)parameters.getNx() * 2. / (T)
        logiT);
442  util::ValueTracer<T> convCheck(1, logVT, epsilon);
443
444  std::cout.precision(std::numeric_limits<T>::digits10);
445  pcout  // << "L0 : " << parameters.getL0() << endl
446        << "Ga : " << parameters.getGa() << endl
447        << "Nx : " << parameters.getNx() << endl
448        << "Tau : " << parameters.getTau() << endl
449        << "Force : " << parameters.getForce() << endl
450        // << "DeltaX : " << parameters.getDeltaX() << endl
451        // << "LatticeNu : " << parameters.getLatticeNu() << endl
452        // << "Force: " << parameters.getForce() << endl
453        << "LatticeU : " << parameters.getLatticeU() << endl
454        //<< "DeltaT : " << parameters.getDeltaT() << endl
455        << "Reseau : " << geoparameters.getReseau() << endl
456        << "VTlog : " << logVT << endl
457        << "cx : " << geoparameters.getCx() << endl;
458  // << "cy : " << geoparameters.getCy() << endl
459  // << "cz " << geoparameters.getCz() << endl
460  // << "poro : " << poro << endl;
461
462  // Creation d'un fichier de sortie permettant d'observer la
        reponse temporelle
463  // du nombre de Darcy
464  char file_output2[39];
465  strcpy(file_output2, dir);
466  strcat(file_output2, "output2.dat");
467  plb_ofstream ofile0(file_output2);
468  ofile0 << std::setprecision(std::numeric_limits<T>::digits10);
469
470  // Initialisation du pas de temps
471  plint iT = 0;
472  // Execution de la boucle principale
473  do {
474      if (iT % logiT == 0) {
475          // Impression du nombre de pas de temps et du nombre de Darcy

```

```

        calcule a
476    // partir du flux massique moyen
477    pcout << "iT : " << iT << "\t Da_avg_m_dot : "
478        << computeAvgMassFlux(lattice) / parameters.getLatticeU
        () << endl;
479
480    // Verification de l'atteinte du regime stationnaire
481    convCheck.takeValue(computeAvgMassFlux(lattice), true);
482    // Impression dans le de l'evolution du nombre de Darcy en
        fonction du
483    // temps
484    ofile0 << iT << " "
485        << computeAvgMassFlux(lattice) / parameters.getLatticeU
        () << endl;
486    }
487
488    // Execution d'un iteration de l'algorithme LBM
489    lattice.collideAndStream();
490    // Incrementation du pas de temps
491    ++iT;
492    }
493    // Verification de la condition de la condition d'atteinte de
        regime
494    // stationnaire, la duree maximale de la simulation, et de la
        stabilite de la
495    // solution
496    while ((convCheck.hasConverged() == 0) && (iT < maxiT) &&
        (getStoredAverageDensity<T>(lattice) > 0));
497
498
499    pcout << "iT : " << iT << endl
500        << "MaxiT : " << maxiT << endl
501        << "ConvCheck : " << convCheck.hasConverged() << endl
502        << "AvgDensity : " << getStoredAverageDensity<T>(lattice) <<
        endl;
503
504    T avg_rho_2 = getStoredAverageDensity<T>(lattice);
505    T Da_avgQ_2 = computeAvgVelComp(lattice) / parameters.getLatticeU
        ();

```

```

506   T Da_avg_m_dot_2 = computeAvgMassFlux(lattice) / parameters.
        getLatticeU();
507   T Force_2 = lattice.getInternalStatistics().getSum(forceIds[0]) /
508       (parameters.getForce() * pow((T)parameters.getNx(),
        3.) * poro);
509   lattice.collideAndStream();
510   lattice.collideAndStream();
511   writeVTK(lattice, parameters, iT);
512
513   // Impression de variables dans les fichiers de sortie
514   if (std::ifstream("output.dat")) {
515   } else {
516       plb_ofstream ofile("output.dat");
517       ofile << "filename L0 Ga Nx Tau Nu F U dXadim dTadim reseau cx
        cy cz R "
518           "poro epsilon iT AvgConvCheck ConvCheck f0 f1 f2 adimF
        AvgDensity "
519           "Da_vel Da_mdot adimF2 AvgDensity2 Da_vel2 Da_mdot2 "
520           "ConvCheckStdDev FluidModel"
521       << endl;
522   }
523   plb_ofstream ofile("output.dat", ios::app);
524   ofile << std::setprecision(std::numeric_limits<T>::digits10) <<
        dir << " "
525       << parameters.getL0() << " " << parameters.getGa() << " "
526       << parameters.getNx() << " " << parameters.getTau() << " "
527       << parameters.getLatticeNu() << " " << parameters.getForce()
        << " "
528       << parameters.getLatticeU() << " " << parameters.getDeltaX()
        << " "
529       << parameters.getDeltaT() << " " << geoparameters.getReseau
        () << " "
530       << geoparameters.getCx() << " " << geoparameters.getCy() <<
        " "
531       << geoparameters.getCz() << " " << poro << " " << epsilon <<
        " " << iT
532       << " " << convCheck.computeAverage() << " " << convCheck.
        hasConverged()

```

```

533     << " " << lattice.getInternalStatistics().getSum(forceIds
        [0]) << " "
534     << lattice.getInternalStatistics().getSum(forceIds[1]) << "
        "
535     << lattice.getInternalStatistics().getSum(forceIds[2]) << "
        "
536     << lattice.getInternalStatistics().getSum(forceIds[0]) /
537         (parameters.getForce() * pow((T)parameters.getNx(),
            3.) * poro)
538     << " " << getStoredAverageDensity<T>(lattice) << " "
539     << computeAvgVelComp(lattice) / parameters.getLatticeU() <<
        " "
540     << computeAvgMassFlux(lattice) / parameters.getLatticeU() <<
        " "
541     << Force_2 << " " << avg_rho_2 << " " << Da_avgQ_2 << " "
542     << Da_avg_m_dot_2 << " "
543     << convCheck.computeStdDev(convCheck.computeAverage()) << "
        " << endl;

```
